Parallel algebraic multilevel Schwarz preconditioners for a class of elliptic PDE systems*

A. Borzì† Valentina De Simone‡ Daniela di Serafino§

October 3, 2013

Abstract

Algebraic multilevel preconditioners for algebraic problems arising from the discretization of a class of systems of coupled elliptic partial differential equations (PDEs) are presented. These preconditioners are based on modifications of Schwarz methods and of the smoothed aggregation technique, where the coarsening strategy and the restriction and prolongation operators are defined using a point-based approach with a primary matrix corresponding to a single PDE. The preconditioners are implemented in a parallel computing framework and are tested on two representative PDE systems. The results of the numerical experiments show the effectiveness and the scalability of the proposed methods. A convergence theory for the twolevel case is presented.

Keywords: systems of elliptic PDEs, algebraic multilevel preconditioners, Schwarz methods, smoothed aggregation, parallel computing.

MSC: 65F08, 65N55, 49J20.

1 Introduction

It is a general consensus that Krylov solvers preconditioned with multilevel Schwarz (MLS) methods [27, 30] are highly efficient and robust iterative schemes for solving linear systems that arise from the discretization of partial differential equation (PDE) problems. MLS methods combine the natural parallelism of the domain decomposition technique, which splits a given problem into a collection of problems of the same type on smaller subdomains, with the optimal convergence property resulting from the application of the multigrid strategy, which exploits a representation of the problem on a hierarchy of discretization meshes. MLS methods can be implemented with the algebraic approach, that is, using only information on the system matrix, without assuming specific knowledge of the discretization used.

*Work partially supported by INdAM-GNCS Projects 'Advanced numerical methods for large-scale nonlinear constrained optimization problems' (2011) and 'Numerical methods and software for preconditioning linear systems arising in PDE and optimization problems' (2012), and by BMBF Verbundproject 05M2013 'ROENOBIO: Robust energy optimization of fermentation processes for the production of biogas and wine'.

†Institut für Mathematik, Universität Würzburg, Campus Hubland Nord, Emil-Fischer-Str. 30, 97074 Würzburg, Germany (alfio.borzi@mathematik.uni-wuerzburg.de).

‡Dipartimento di Matematica e Fisica, Seconda Università degli Studi di Napoli, viale A. Lincoln 5, 81100 Caserta, Italy (valentina.desimone@unina2.it).

§Dipartimento di Matematica e Fisica, Seconda Università degli Studi di Napoli, viale A. Lincoln 5, 81100 Caserta, Italy, and Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR), CNR, Napoli, Italy (daniela.diserafino@unina2.it).

In the case of scalar PDEs, the development of algebraic MLS methods follows canonical guidelines: the subproblems in one-level Schwarz methods are usually obtained by building an overlapping partition of the adjacency graph of the matrix [7], while the coarse-level correction is obtained by applying typical coarsening strategies of algebraic multigrid (AMG) methods; see, e.g., [15, 29]. However, many application problems require to solve systems of PDEs and an effective extension of algebraic MLS methods is not straightforward. It is the purpose of this paper to present and investigate an algebraic MLS strategy for a class of systems of elliptic PDEs. This class is defined in terms of the structure of the system and of the properties of the single elliptic equations, therefore we do not claim that our approach is purely algebraic although only general information is required.

A reasonable step towards a successful MLS procedure for PDE systems is the use of a "point-based" approach, where the unknown variables that may correspond to a grid point are treated simultaneously (see, e.g., [18, 20, 22, 26]). In this case, one usually reorders the unknowns so that those corresponding to the same grid point are consecutive, thus obtaining a block-sparse matrix of coefficients where the blocks account for the contribution of all the equations to a single grid point and the block structure provides a compact information on the overall problem. As noted in [18], the point-based approach can be applied also in a completely algebraic context where no physical grid points are available, provided that we are able to identify blocks of variables that can be treated as a whole.

In this context, a fundamental issue is the definition of connectivity between blocks of variables, which is the basis for the coarsening strategy [29]. In the scalar case, the connectivity among neighboring variables is given by the stencil of the discretized differential operator, that is naturally reflected in the structure and the entries of the matrix. In the case of PDE systems, the point-based view requires a generalization of the connectivity concept to blocks of variables resulting from the presence of multiple PDE operators. The overlapping block-partitions used by the Schwarz methods are usually built by reasoning as in the scalar case, but on the graph obtained by considering the matrix blocks as points. However, the generalization of the coarsening process is more critical, since the strength of the block connections must be defined. Some strategies for block coarsening are outlined in [18]. However, as noted in [2], the transfer operators resulting from the point-based strategy might significantly depend not only on the differential part, but also on the zero-order terms in the equations, thus resulting in a possible loss of efficiency of the multilevel method. On the other hand, the effective geometric multigrid practice [3] is to neglect the non-differential part. For this reason, we aim at developing an algebraic coarsening that closely resembles the geometric approach.

In this paper, we consider a class of elliptic PDE systems of the following type

$$-\sum_{k=1}^{d} \frac{\partial}{\partial x_k}\left(d_k^{(i)} \frac{\partial u^{(i)}}{\partial x_k}\right) + \sum_{m=1}^{n_e} b^{(i,m)} u^{(m)} = f^{(i)}, \quad i = 1, \ldots, n_e, \tag{1}$$

where $\Omega \subset \mathbb{R}^d$ and $d_k^{(i)}, b^{(i,m)} \in L^\infty(\Omega)$ and $f^{(i)} \in L^2(\Omega)$ are given. We note that the coupling of different unknowns takes place only through the zero-order terms. On the boundary $\partial\Omega$, the following Robin boundary conditions are assigned

$$\alpha \frac{\partial u^{(i)}}{\partial \mathbf{n}} + \beta\, u^{(i)} = \gamma^{(i)}, \quad i = 1, \ldots, n_e, \tag{2}$$

where $\alpha, \beta, \in \mathbb{R}$, $\gamma^{(i)} \in L^2(\Omega)$ and $\mathbf{n}$ is the outward normal to $\partial\Omega$. The formulation (1)-(2) covers a wide class of applications, including, e.g., optimality systems of some PDE-constrained optimization problems and diffusion-reaction systems. For our purpose, we require that the second-order operators have similar structure in terms of anisotropy.

We analyse a point-based algebraic approach for building MLS preconditioners where the restriction and prolongation operators associated with both the Schwarz matrix partitioning and the coarsening strategy are defined by using a primary matrix that results from the discretization of the second-order differential operator corresponding to a single equation in the PDE system and neglecting the zero-order coupling terms. In particular, we use this primary matrix in the context of the smoothed aggregation coarsening technique [4, 5, 32]. Furthermore, we present a parallel implementation of the proposed preconditioners in the framework of MLD2P4 (Multilevel Domain Decomposition Parallel Preconditioners Package based on PSBLAS) [11]. We remark that our approach has the great advantage of requiring a setup process whose computational and memory complexity is similar to that corresponding to a single PDE.

We notice that the point-based single-operator approach has already been exploited in [2], in the context of (sequential) AMG methods with Stüben-type coarsening. Our aim is to extend and assess this methodology in the case of parallel algebraic MLS methods based on smoothed aggregation. We show that our MLS preconditioners are able to achieve the effectiveness of the scalar case as well as to yield good parallel performance. They also appear to be robust with respect to problem parameters such as the regularization parameter in elliptic PDE-constrained optimization problems. As expected, the effectiveness of the preconditioners worsen on systems of PDEs where the elliptic operators show strong anisotropies in different directions; nevertheless, advantageous computational performance is achieved as long as the differences in the anisotropies are moderate. Along this line, we remark that our numerical experience and results reported in the mathematical literature demonstrate that the use of Krylov iterations in combinations with MLS preconditioners provides a robust alternative to the usage of a pure multigrid scheme. In fact, in the latter case more sophisticated 'ad-hoc' coarsening techniques and intergrid transfer operators generally need to be constructed to guarantee an efficient multilevel method.

This paper is organized as follows. In Section 2 we provide representative elliptic PDE systems that will be used to test our MLS approach. In Section 3 we describe our MLS preconditioners. In Section 4 we present their parallel implementation. In Section 5 we report the results of numerical experiments aimed at analyzing the effectiveness and the parallel performance of the preconditioners on the selected test problems. In Section 6 we provide theoretical converge results for a twolevel case. A section of conclusions completes this work.

2 Elliptic PDE systems

An important class of elliptic PDE problems of type (1)-(2) is given by the optimality systems arising from a variety of elliptic PDE-constrained optimization problems [23]. In particular, we consider the following distributed optimal control problem

$$\begin{array}{ll} \min_{u\in L^2(\Omega)} & \frac{1}{2}\|y-z\|^2_{L^2(\Omega)} + \frac{\nu}{2}\|u\|^2_{L^2(\Omega)}, \\ \text{s.t.} & -\Delta y - u = g \quad \text{in } \Omega, \\ & \qquad y = 0 \quad \text{on } \partial\Omega, \end{array} \tag{3}$$

where $\Omega \subset \mathbb{R}^d$ is convex, and $g, z \in L^2(\Omega)$ are given functions. In this context, y and z are the so-called state and target functions, respectively. The function $u \in L^2(\Omega)$ represents the control and the regularization parameter $\nu > 0$ is the weight of the cost of the control. For $d = 2$, problem (3) may describe the problem of controlling the temperature distribution y of a material plate, kept equal to zero along the boundary, through the application of a source u

aimed at getting a target temperature profile. The optimal solution of (3) is characterized by the following optimality conditions:

$$\begin{aligned} -\Delta y - u &= g \quad \text{in } \Omega, \\ -\Delta p + y &= z \quad \text{in } \Omega, \\ \nu u - p &= 0 \quad \text{in } \Omega, \\ y = 0,\ p &= 0 \quad \text{on } \partial\Omega. \end{aligned}$$

However, we can use the condition $\nu u - p = 0$ to reduce the system as follows:

$$\begin{aligned} -\nu\,\Delta y - p &= \nu\, g \quad \text{in } \Omega, \\ -\Delta p + y &= z \quad \text{in } \Omega, \\ y = 0,\ p &= 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{4}$$

This is one of the test problems used to analyse the behavior of the MLS preconditioners presented in this paper (details about it are given in Section 5.1). Notice that the ill-conditioning of this problem increases as ν gets smaller.

Another wide class of differential problems of the type (1)-(2) is given by steady-state linear diffusion-reaction problems that describe a variety of applications, including chemical and biological processes [25]. In particular, we consider diffusion-reaction problems where some diffusion operators are anisotropic along different directions as follows:

$$\begin{aligned} -\frac{\partial^2 u}{\partial x_1^2} - \frac{\partial^2 u}{\partial x_2^2} + u + v - w &= f \quad \text{in } \Omega, \\ -\varepsilon\frac{\partial^2 v}{\partial x_1^2} - \frac{\partial^2 v}{\partial x_2^2} + u + v - w &= g \quad \text{in } \Omega, \\ -\frac{\partial^2 w}{\partial x_1^2} - \varepsilon\frac{\partial^2 w}{\partial x_2^2} - u - v + w &= h \quad \text{in } \Omega, \\ u = v = w &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{5}$$

where $0 < \epsilon < 1$ and $f, g, h \in L^2(\Omega)$. This problem is significant in the context of this work, since, as ε decreases, it becomes a more difficult test case for MLS methods because of the different anisotropies.

3 Multilevel Schwarz preconditioners for systems of elliptic PDEs

In this section, we present our approach for building MLS preconditioners for linear systems arising from the discretization of elliptic differential problems of the type (1)-(2). After discretization, these problems result in an algebraic system of the following form

$$A_l^{(i)} u_l^{(i)} + \sum_{m=1}^{n_e} B_l^{(i,m)} u_l^{(m)} = f_l^{(i)}, \quad i = 1, \ldots, n_e, \tag{6}$$

where $A_l^{(i)}$ is the discrete operator corresponding to the differential part of the ith PDE and $B_l^{(i,m)}$ represents the zero-order term associated with the unknown vector $u_l^{(m)}$ in that PDE. We

assume that $A_l^{(i)}$ and the right-hand side $f_l^{(i)}$ take into account also the discretization of the boundary conditions. We remark that the structure (6) identifies the class of problems that we are considering. As we discuss next, MLS algorithms construct a hierarchy of coarser systems of the type (6); the index $l = 0, \ldots, L$ in (6) refers to the lth coarsening level, where L identifies the original (finest) system and 0 identifies the coarsest system.

System (6) can be also represented in a more compact form as follows

$$\mathcal{A}_l \mathbf{u}_l = \mathbf{f}_l, \tag{7}$$

where

$$\begin{gathered} \mathcal{A}_l = \hat{\mathcal{A}}_l + \mathcal{B}_l, \\ \hat{\mathcal{A}}_l = diag\left(A_l^{(1)}, \ldots, A_l^{(n_e)}\right), \quad \mathcal{B}_l = \left(B_l^{(i,m)}\right)_{i,m=1,\ldots,n_e}, \\ \mathbf{u}_l = \left(u_l^{(i)}\right)_{i=1,\ldots,n_e}, \quad \mathbf{f}_l = \left(f_l^{(i)}\right)_{i=1,\ldots,n_e}. \end{gathered} \tag{8}$$

The total number of grid points at level l is denoted by N_l and the total number of unknowns in (7) by $\bar{N}_l := n_e N_l$; furthermore, the set of row indices of the matrix $\mathcal{A}_l$ is denoted by $\mathcal{V}_l$. Note that, even if we use a point-based approach, we describe our method without reordering the rows and columns of system (7) to have blocks of coefficients associated with grid points. This is a consequence of our choice of the primary matrix in the point-based approach, as shown next.

We consider algebraic one-level additive Schwarz (AS) methods and recall how they work in a general scalar algebraic approach (see [7] for details). In this case, one starts from a 0-overlap partition of the index set $\mathcal{V}_l$, i.e., from a set of t_l disjoint nonempty sets $\mathcal{V}_{l,j}^0 \subset \mathcal{V}_l$ such that $\cup_{j=1}^{t_l} \mathcal{V}_{l,j}^0 = \mathcal{V}_j$, and generate overlapping partitions of $\mathcal{V}_l$ by using the adjacency graph of the matrix $\mathcal{A}_l$. For $\delta > 0$, a δ-overlap partition of $\mathcal{V}_l$ is constructed recursively, by considering the sets $\mathcal{V}_{l,j}^\delta \supset \mathcal{V}_{l,j}^{\delta-1}$ obtained by including the vertices that are adjacent to any vertex in $\mathcal{V}_{l,j}^{\delta-1}$. Once a δ-overlap partition of $\mathcal{V}_l$ has been built, we can associate with each of its sets $\mathcal{V}_{l,j}$ the restriction operator $\mathcal{R}_{l,j}$ that maps a vector $\mathbf{u}_l$ of size $\bar{N}_l$ onto the vector made only of the entries of $\mathbf{u}_l$ with indices in $\mathcal{V}_{l,j}$, and the prolongation operator $\mathcal{P}_{l,j} = \mathcal{R}_{l,j}^T$. These are the basic ingredients to obtain Schwarz preconditioners.

In our approach, we define the overlapping partitions and the corresponding restriction and prolongation operators by applying the primary-matrix approach outlined in the introduction, i.e., by using a single block $A_l^{(j)}$ in $\hat{\mathcal{A}}_l$, which is assumed to be representative of all the blocks. The choice of the primary matrix is a delicate issue. Our numerical experience and previous work on the ellipticity symbol of PDE systems [34,35] suggest that the equation with better (in the sense of more uniform) ellipticity should provide the primary matrix. On the other hand, if the discretized differential operators $A_l^{(i)}$ have similar features, any of them can be chosen as the primary matrix. For the two problems described in Section 2, the primary matrix is chosen according to these guidelines (see Sections 5.1 and 5.2). In the following, to simplify the notation, we denote by A_l the block selected as primary matrix.

We use the strategy previously described to build a δ-overlap partition of the index set V_l associated with A_l, $\left\{V_{l,j},\ j = 1, \ldots, t_l\right\}$, and the associated restriction and prolongation operators, $R_{l,j}$ and $P_{l,j}$, respectively. To build Schwarz methods for the whole system (7), we extend $R_{l,j}$ and $P_{l,j}$ as follows

$$\hat{\mathcal{P}}_{l,j} = I_{n_e} \otimes P_{l,j}, \quad \hat{\mathcal{R}}_{l,j} = I_{n_e} \otimes R_{l,j}, \tag{9}$$

where $\otimes$ is the Kronecker product and I_{n_e} is the identity matrix of dimension n_e. The operators (9) are used as in classical Schwarz methods to build the following matrices:

$$\mathcal{A}_{l,j} = \hat{\mathcal{R}}_{l,j}\mathcal{A}_l\hat{\mathcal{P}}_{l,j}. \tag{10}$$

This procedure naturally associates a row-index set $\mathcal{V}_{l,j}$ with each matrix $\mathcal{A}_{l,j}$.

We define our AS preconditioner as follows

$$\mathcal{M}_l^{AS} = \sum_{j=1}^{t_l} \hat{\mathcal{P}}_{l,j}\mathcal{A}_{l,j}^{-1}\hat{\mathcal{R}}_{l,j}, \tag{11}$$

where $\mathcal{A}_{l,j}$ is supposed to be nonsingular. An approximate inverse $\mathcal{S}_{l,j}$ of $\mathcal{A}_{l,j}$ is usually considered instead of $\mathcal{A}_{l,j}^{-1}$, to save computational cost. Variants of this preconditioner are obtained by neglecting the overlap when the restriction or the prolongation operators are applied, as in the scalar case.

To build the coarse index set and the relevant transfer operators, we use the smoothed aggregation technique [4, 5, 32]; however, we do not apply the block approach proposed in [32], where the connectivity among blocks of variables is measured using the norms of the matrix blocks resulting from the point-based approach. Instead, we build the coarse index set and the prolongation and restriction operators by using only the selected matrix A_l, which is our primary matrix. The indices in V_l are grouped into subsets, forming a disjoint covering $\{C_{l,q},\ q = 1,\ldots,N_{l-1}\}$ of V_l. Each subset $C_{l,q}$, contains indices strongly coupled to a certain "root" index in V_l; two indices r and s are said strongly coupled if the following holds:

$$|a_{rs}^l| \geq \vartheta\sqrt{|a_{rr}^l a_{ss}^l|},$$

where a_{rs}^l denotes the (r,s)th entry of the matrix A_l and $\vartheta > 0$ is a suitably chosen threshold [32]. A tentative prolongator $\tilde{P}_{l-1}$, mapping a vector of length N_{l-1} onto a vector of length N_l, is built by exploiting information on the near null space of A_l [24]; for example, if A_l stems from the discretization of the negative Laplacian, $\tilde{P}_{l-1}$ has the following form

$$\tilde{P}_{l-1} = (\tilde{p}_{sq}^{l-1}), \quad \tilde{p}_{sq}^{l-1} = \begin{cases} 1 & \text{if } s \in C_{l,q}, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

To reduce the energy of the prolongator columns, a damped Jacobi smoother is applied to $\tilde{P}_{l-1}$, thus obtaining

$$P_{l-1} = (I_{N_l} - \omega_l D_l^{-1} A_l)\tilde{P}_{l-1},$$

where I_{N_l} is the identity matrix of dimension N_l, D_l is the diagonal part of A_l, $\omega_l = 4/(3\rho_l)$ and ρ_l is an upper bound on the spectral radius of $D_l^{-1}A_l$ [4]. To obtain a smoothed prolongator $\mathcal{P}_{l-1}$ for system (7), the operator P_{l-1} is extended as follows:

$$\hat{\mathcal{P}}_{l-1} = I_{n_e} \otimes P_{l-1}. \tag{13}$$

The coarse-level matrix $\mathcal{A}_{l-1}$ is computed as

$$\mathcal{A}_{l-1} = \hat{\mathcal{R}}_{l-1}\mathcal{A}_l\hat{\mathcal{P}}_{l-1}, \tag{14}$$

where $\hat{\mathcal{R}}_{l-1} = \hat{\mathcal{P}}_{l-1}^T$. We observe that $\mathcal{A}_{l-1}$, $\hat{\mathcal{R}}_{l-1}$ and $\hat{\mathcal{P}}_{l-1}$ identify the associated index space, $\mathcal{V}_{l-1}$; furthermore, $\mathcal{A}_{l-1}$ has the same block structure as $\mathcal{A}_l$, hence the strategy described so far can be applied at any level l.

We note that, by construction, the restriction and prolongation operators, and therefore the coarsening, are independent of the zero-order terms in the PDEs. On the other hand, to take into account the coupling introduced by the matrix $\mathcal{B}_l$, the matrix $\mathcal{A}_{l-1}$ in (14) as well as the matrices $\mathcal{A}_{l,j}$ in (10) are computed by applying the relevant prolongation and restriction operators to the matrix $\mathcal{A}_l$ and not to $\hat{\mathcal{A}}_l$. Therefore, using as primary matrix the $N_l \times N_l$ block A_l reduces the computational cost for the coarsening and the construction of the restriction and prolongation operators with respect to the classical block approach. On the other hand, the primary matrix $A_{l-1} = R_{l-1}A_lP_{l-1}$, to be used at the next coarse level, must be computed too, but the cost of this operation is generally paid off by the smaller cost of the coarsening and the transfer operators, as long as the number n_e of PDEs is large enough.

The hierarchy of coarse matrices and the inter-level transfer operators may be combined in several ways with the AS preconditioners at each level, to obtain different multilevel preconditioners (see, e.g., [1, 27]). The simplest combination is the additive scheme, where the coarse-level correction is added to the AS preconditioner at each level; if two levels l and $l-1$ are considered, the additive multilevel Schwarz (AMLS) preconditioner takes the following form:

$$\mathcal{M}_l^{AMLS} = \hat{\mathcal{P}}_{l-1}\mathcal{A}_{l-1}^{-1}\hat{\mathcal{R}}_{l-1} + \sum_{j=1}^{t_l} \hat{\mathcal{P}}_{l,j}\mathcal{A}_{l,j}^{-1}\hat{\mathcal{R}}_{l,j}.$$

Letting

$$\begin{aligned}
&\mathcal{P}_{L,j} = \hat{\mathcal{P}}_{L,j}, && j = 1,\ldots,t_L,\\
&\mathcal{P}_l = \hat{\mathcal{P}}_L\hat{\mathcal{P}}_{L-1}\cdots\hat{\mathcal{P}}_l, && l = 0,\ldots,L-1,\\
&\mathcal{P}_{l,j} = \mathcal{P}_l\hat{\mathcal{P}}_{l,j}, && l = 1,\ldots,L-1,\ \ j = 1,\ldots,t_l,\\
&\mathcal{P}_{0,j} = \mathcal{P}_0, && j = t_0 = 1,\\
&\mathcal{R}_{l,j} = \mathcal{P}_{l,j}^T, && l = 0,\ldots,L,\ \ j = 1,\ldots,t_l,
\end{aligned} \tag{15}$$

it is easy to verify that the additive multilevel Schwarz preconditioner using the $L+1$ levels $l = 0,\ldots,L$ (where L and 0 denote the finest and the coarsest level, respectively) can be written as follows [27]:

$$\mathcal{M}^{AMLS} = \sum_{l=0}^{L}\sum_{j=1}^{t_l} \mathcal{P}_{l,j}\mathcal{A}_{l,j}^{-1}\mathcal{R}_{l,j}; \tag{16}$$

this matrix form will be used in Section 6 to prove some theoretical results.

Multiplicative combinations of the coarse matrices and the transfer operators with the AS preconditioners may be also considered (see, e.g., [27] for details); the resulting preconditioners are called hybrid, since they are additive at each level l and multiplicative in passing from level l to level $l-1$ and vice versa. An example is given by the symmetrized multiplicative multilevel Schwarz (SMMLS) preconditioner [27, Algorithm 3.2.2], that we use in our numerical experiments in Section 5. For simplicity, we illustrate it in Figure 1 instead of providing its matrix form. Notice that the SMMLS scheme is equivalent to a V-cycle with zero starting guess at the finest level; furthermore, this preconditioner is symmetric if the matrix $\mathcal{A}_l$ is symmetric.

4 Parallel implementation issues

The MLS preconditioners described in the previous sections were implemented in the framework of MLD2P4, a Fortran 95 package of parallel algebraic multilevel Schwarz preconditioners based

SMMLS$(l, \mathcal{A}_l, \mathbf{f}_l)$
if $(l \neq 0)$ then
  $\mathbf{u}_l = \mathcal{M}_l^{AS}\mathbf{f}_l$
  $\mathbf{f}_{l-1} = \hat{\mathcal{R}}_{l-1}(\mathbf{f}_l - \mathcal{A}_l\mathbf{u}_l)$
  $\mathbf{u}_{l-1} =$ **SMMLS**$(l-1, \mathcal{A}_{l-1}, \mathbf{f}_{l-1})$
  $\mathbf{u}_l = \mathbf{u}_l + \hat{\mathcal{P}}_{l-1}\mathbf{u}_{l-1}$
  $\mathbf{u}_l = \mathbf{u}_l + \mathcal{M}_l^{AS}(\mathbf{f}_l - \mathcal{A}_l\mathbf{u}_l)$
else
  $\mathbf{u}_l = \mathcal{A}_l^{-1}\mathbf{f}_l$
endif
return $\mathbf{u}_l$

Figure 1: The symmetrized multiplicative multilevel Schwarz method.

on smoothed aggregation, designed for distributed-memory architectures [11]. MLD2P4 has a layered software architecture, which is the result of an object-based modular approach reflecting the intrinsic hierarchy of multilevel methods. In this context, "objects" of increasing complexity are obtained by suitably combining simpler objects, that are general enough to ensure flexibility and reuse, while meeting performance requirements. This approach allows the introduction of new features into the package, as well as the modification of existing ones. At the bottom of this architecture, the PSBLAS library [16,17] provides parallel sparse linear algebra computation and communication kernels used as basic building blocks in the preconditioner setup and application (see [9,11] for details). These kernels are implemented on the top of BLACS [13] and MPI [28], which are "de-facto" standard message-passing environments. The preconditioners available in MLD2P4 can be used with the Krylov solvers implemented in PSBLAS.

While a detailed description of the parallel implementation of our MLS preconditioners is beyond the scope of this paper, here we outline very general implementation choices, to better understand the parallel performance results reported in the next section. The preconditioner data structure of MLD2P4 assume a row-block distribution of the input matrix, where the rows may also be non-contiguous, and a consistent distribution of the associated vectors of unknowns and right-hand sides; the number of row-blocks equals the number of available processors. This distribution naturally induces a partitioning of the row-index set associated with the matrix, which is the basis for building the overlapping partitions used by the Schwarz methods. To keep the same data partitioning in all the phases of the construction and application of our MLS preconditioners, we consider a point-based reordering of (7), where the unknowns associated with the same grid point are consecutive. We denote the resulting system as follows

$$\bar{\mathcal{A}}_l\bar{\mathbf{u}}_l = \bar{\mathbf{f}}_l, \tag{17}$$

with

$$\bar{\mathcal{A}}_l = \Pi\mathcal{A}_l\Pi^T, \quad \bar{\mathbf{u}}_l = \Pi^T\mathbf{u}, \quad \bar{\mathbf{f}}_l = \Pi\mathbf{f}_l,$$

where Π is a suitable permutation matrix. Note that $\bar{\mathcal{A}}_l$, $\bar{\mathbf{u}}_l$ and $\bar{\mathbf{f}}_l$ have a block structure, where

each block accounts for all the variables associated with a grid point. We have

$$\bar{\mathcal{A}}_l = \left(\bar{A}_l^{(r,s)} + \bar{B}_l^{(r,s)}\right)_{r,s=1,\ldots,N_l}, \quad \bar{A}_l^{(r,s)}, \bar{B}_l^{(r,s)} \in \mathbb{R}^{n_e \times n_e},$$

$$\bar{\mathbf{u}}_l = \left(\bar{u}_l^{(r)}\right)_{r=1,\ldots,N_l}, \qquad \bar{u}_l^{(r)} \in \mathbb{R}^{n_e},$$

$$\bar{\mathbf{f}}_l = \left(\bar{f}_l^{(r)}\right)_{r=1,\ldots,N_l}, \qquad \bar{f}_l^{(r)} \in \mathbb{R}^{n_e}.$$

The primary matrix $A_l = A_l^{(j)}$ is obtained from $\bar{\mathcal{A}}_l$ by considering the jth row and column of each block $\bar{A}_l^{(r,s)}$. Notice that, even if we do not assume any specific knowledge of the discretization scheme applied to the PDE problem, we must be able to distinguish between the part of the matrix accounting for the discretization of the differential terms and the part of the matrix accounting for the zero-order terms. In our opinion, this is not a strong requirement when solving algebraic problems arising from the discretization of a system of PDEs.

A row-block decomposition of system (17) is considered, where all the rows associated with the same grid point are assigned to the same processor; this naturally results in a partition of the primary matrix A_l and the corresponding row-index set V_l. This is the 0-overlap partition used by Schwarz methods to build a δ-overlap partition of V_l and its extension to the whole index set $\mathcal{V}_l$. The implementation of our one-level Schwarz methods is accomplished by modifying the MLD2P4 components that identify the overlap indices and retrieve the corresponding matrix rows on each processor. Concerning the coarsening strategy, MLD2P4 implements a decoupled aggregation algorithm, that acts locally on the subset of indices assigned to each processor, neglecting any connection with indices held by other processors. This algorithm does not require communication among the processors, thus allowing a significant time saving with respect to other parallel aggregation algorithms. On the other hand, it may generate nonuniform aggregates near the boundary indices (the indices held by a processor that are adjacent, in the matrix graph, to indices in other processors) and is dependent on the number of processors and on the initial partitioning of the matrix. Nevertheless, this approach has shown to be able to produce good results in practice [6, 9, 10, 31]. In our case, the decoupled aggregation is performed on V_l and then the resulting aggregates and prolongation operators are extended according to the point-based ordering of the unknowns. The local indices to be aggregated on each processor are determined by the initial 0-overlap partition of V_l. The aggregation and the construction of the smoothed restriction and prolongation operators in our MLS preconditioners were obtained by suitably modifying the corresponding tasks implemented in MLD2P4, to deal with the primary matrix instead of the whole matrix. We note that the modifications introduced in MLD2P4 are implemented by using the data structures of MLD2P4 and the basic sparse linear algebra computation and communication kernels provided by PSBLAS, to preserve the objectives of efficiency, portability, and general applicability of the original package.

5 Numerical experiments

We performed numerical experiments to evaluate the effectiveness of our parallel point-based algebraic MLS preconditioners. We applied the preconditioners to solve algebraic problems arising from the discretization of the optimality system (4) and the anisotropic diffusion-reaction problem (5). These PDE problems are defined on the domain $\Omega = [0,1] \times [0,1]$ and are approximated by second-order central finite-differences on a uniform $n \times n$ grid. Henceforth, the corresponding linear systems are referred to as problem 1 and problem 2, respectively. Notice

that the former is nonsymmetric, while the latter is symmetric positive definite and anisotropic. For each problem we considered different values of the parameters and different sizes of the discretization grids.

We report results obtained using the SMMLS scheme with different levels. The coarsest matrix $\mathcal{A}_l$ was replicated on the processors and the corresponding system was solved by the sparse LU factorization implemented in UMFPACK [12], available through its MLD2P4 interface. According to [32], the threshold ϑ in the aggregation algorithm was chosen adaptively as $\vartheta = 0.08 \cdot 1/2^{L-l}$, where l is the current level. The block-Jacobi (BJAC) method was used as basic Schwarz preconditioner, with the $ILU(0)$ factorization to approximate the blocks $\mathcal{A}_{l,j}^{-1}$ in (11); notice that $ILU(0)$ reduces to an incomplete LDL^T factorization when $\mathcal{A}_{l,j}$ is symmetric positive definite.

The preconditioners were coupled with Krylov solvers from PSBLAS, i.e., the BiCGStab method was applied to problem 1 and the Conjugate Gradient (CG) method to problem 2. The zero vector was used as initial guess. The iterations were stopped when the 2-norm of the relative residual was smaller than a fixed tolerance, set to 10^{-6}. Furthermore, a maximum number of 1000 iterations was considered. In the following, the SMMLS preconditioners are denoted by xLEV, where x is the number of levels.

For comparison purposes, we also considered the classical block version of the SMMLS preconditioners based on the smoothed aggregation technique, as implemented in the Trilinos/ML package [19]. All the components and parameters of the Trilinos/ML preconditioners were chosen as in our preconditioners, except the threshold ϑ in the aggregation algorithm, which was set to the same value at all the levels because ML does not allow a variable choice of it (see Sections 5.1 and 5.2 for more details). The ML preconditioners were coupled with the BiCGStab and CG implementations provided by AztecOO in the context of the Trilinos framework [21], using the zero initial guess and the stopping criterion based on the 2-norm of the relative residual, with tolerance 10^{-6} and maximum number of iterations equal to 200.

All the experiments were carried out on a HP XC 6000 Linux cluster available at ICAR-CNR, Naples branch, Italy. Each node of the cluster comprises an Intel Itanium 2 Madison bi-processor, with clock frequency of 1.4 Ghz, and 4 GB of RAM; it runs HP Linux for High Performance Computing 3 (kernel 2.4.21). The interconnection network is Quadrics QsNetII Elan 4, which has a sustained bandwidth of 900 MB/sec. and a latency of about 5 μsec. for large messages. Our preconditioners were implemented within MLD2P4 1.2.1, installed on the top of PSBLAS 2.3.4, ATLAS 3.6.0, BLACS 1.1 and the HP MPI implementation 2.01. UMFPACK 4.4 was used with MLD2P4. The software was compiled with the version 4.3 of the GNU Compiler Collection. The tests were performed in double precision; the execution times were measured using the PSBLAS function *psb_wtime*. We used Trilinos 8.0.5, installed in the same environment.

5.1 PDE-constrained optimization problem

In the PDE-constrained optimization problem (3), the target function z and the right-hand side g were chosen as follows:

$$z(x,y) = \sin(2\pi x)\cos(2\pi y), \quad g(x,y) = \sin(2\pi x)\sin(2\pi y).$$

The preconditioners were built using as a primary matrix the one arising from the discretization of the negative Laplacian and were tested varying the regularization parameter, ν, the number of grid points at the finest level, $N_L = n^2$, and the number of processors, np. Note that with this choice of the primary matrix the multilevel prolongation and restriction operators are independent of ν.

np	BJAC	2LEV	3LEV	4LEV	5LEV
$\nu = 10^{-3}$					
1	526	4	6	7	9
2	800	4	5	7	9
4	588	4	6	8	10
8	643	4	6	7	10
16	602	4	5	7	10
32	830	5	6	8	11
$\nu = 10^{-5}$					
1	—	4	7	8	11
2	—	5	7	8	13
4	—	5	6	9	13
8	—	5	7	9	10
16	—	5	7	9	12
32	—	5	7	9	14
$\nu = 10^{-7}$					
1	—	5	8	9	12
2	—	5	7	9	11
4	—	5	6	9	9
8	—	5	7	9	9
16	—	5	6	8	10
32	—	5	7	9	11

Table 1: Problem 1 - number of iterations as ν varies, for $n = 1000$ (— indicates that 1000 iterations were performed without satisfying the accuracy requirement).

np	2LEV	3LEV	4LEV	5LEV
$n = 500$				
1	4	5	7	8
2	4	6	7	9
4	4	6	7	8
8	4	5	7	10
16	4	6	7	12
32	5	6	7	12

Table 2: Problem 1 - number of iterations for $n = 500$ and $\nu = 10^{-3}$.

Table 1 shows the number of BiCGStab iterations for $\nu = 10^{-3}, 10^{-5}, 10^{-7}$ and $n = 1000$. This grid size correspond to a linear system with 3 million unknowns. We see that, with all the levels, the number of iterations is almost constant with respect to ν and np. Notice that the BJAC preconditioner leads to convergence only with $\nu = 10^{-3}$. We remark that results of further experiments show that our multilevel preconditioners provide the same performance when higher accuracy is required. In particular, the number of iterations with tolerance 10^{-12} is lower than twice the number of iterations obtained with tolerance 10^{-6}.

We also report, in Table 2, the iterations obtained by the multilevel preconditioners with grid size $n = 500$, in the case $\nu = 10^{-3}$. By comparing these data with the corresponding ones for $n = 1000$, we see that the number of iterations is almost independent of the grid size. The same behaviour is obtained with $\nu = 10^{-5}, 10^{-7}$. Thus, the preconditioners exhibit the typical behaviour of multilevel Schwarz methods for scalar problems.

Next, we analyse the strong scaling of our MLS preconditioners. In Figure 2 we depict the execution times and the speedup values obtained with the different preconditioners for $n = 1000$

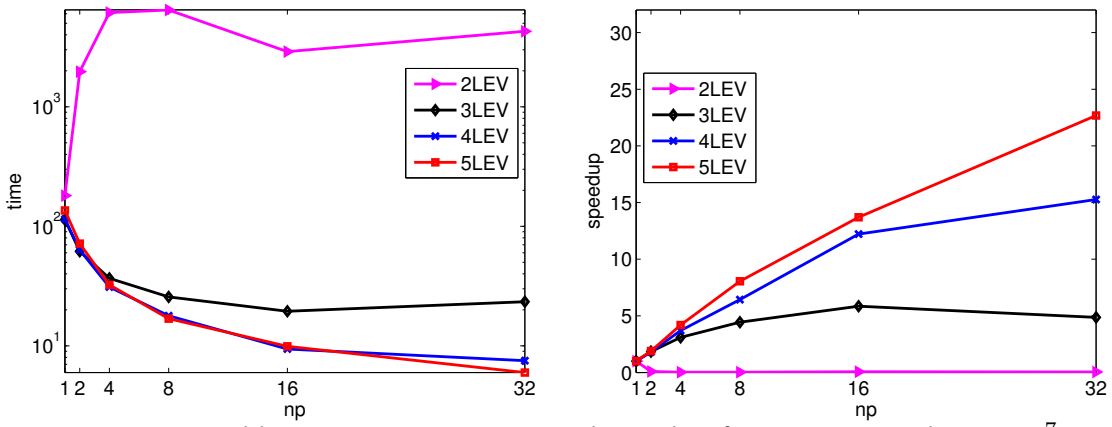

Figure 2: Problem 1 - execution time and speedup for $n = 1000$ and $\nu = 10^{-7}$.

ν	2LEV	3LEV	4LEV
Classical			
10^{-3}	4	5	5
10^{-5}	5	81	—
10^{-7}	5	—	—
New			
10^{-3}	4	5	7
10^{-5}	4	7	8
10^{-7}	5	7	7

Table 3: Problem 1 - number of iterations obtained on one processor with the classical block version (Classical) and our version (New) of SMMLS, for $n = 500$ and $\vartheta = 10^{-2}$ (— indicates that 200 iterations were performed without satisfying the accuracy requirement).

and $\nu = 10^{-7}$ (similar values were obtained with $\nu = 10^{-3}, 10^{-5}$). The CPU times include both the setup of the preconditioner and the solution of the linear system. We note that, on a single processor, larger grid sizes (e.g., $n = 2000$) resulted in too high memory requirements, yielding either out-of-memory faults in UMFPACK or huge execution times, which do not allow a fair evaluation of strong scaling. Therefore, we do not consider larger problems here. Figure 2 shows that 5LEV achieves the best performance in terms of both time and speedup; in particular, the speedup on 32 processors is about 23, which can be considered satisfactory. As expected, 2LEV results much more expensive than the other preconditioners, because of the cost of replicating, factorizing and solving a large coarse-level system. It also appears that the BJAC preconditioner is much less efficient than the multilevel preconditioners with at least 3 levels.

In another series of experiments, we compared our preconditioners with the corresponding Trilinos/ML preconditioners, implementing the classical block approach to systems of PDEs. The experiments were performed using a single processor, since we are interested in evaluating our coarsening approach against the classical one rather than our parallel implementation against the Trilinos one. We considered $n = 500$, from 2 to 4 levels, and different values of the aggregation threshold, i.e., $\vartheta = 0, 10^{-1}, 10^{-2}, 10^{-3}$, for both Trilinos and our preconditioners, in order to make a fair comparison. In Table 3 we report results obtained with $\vartheta = 10^{-2}$; the behaviour of the preconditioners with the other values of ϑ is similar. We see that with $\nu = 10^{-3}$ the

np	BJAC	3LEV	4LEV	5LEV
$\varepsilon = 1$				
1	739	8	10	12
2	874	8	10	13
4	834	8	10	13
8	837	8	11	13
16	886	8	10	15
32	860	9	11	16
$\varepsilon = 10^{-2}$				
1	712	35	41	42
2	809	36	42	43
4	815	35	42	43
8	876	35	42	42
16	920	35	43	50
32	—	36	44	71
$\varepsilon = 10^{-4}$				
1	651	24	24	24
2	681	73	74	74
4	710	76	77	78
8	776	92	91	92
16	865	139	140	140
32	976	178	181	183

Table 4: Problem 2 - number of iterations as ε varies, for $n = 1000$ (— indicates that *maxit* was achieved without satisfying the accuracy requirement).

two preconditioners are comparable. However, as the value of ν decreases, the classical block preconditioner loses its effectiveness when more than two levels are used, and generally does not lead to convergence, while our preconditioners lead to convergence with a modest number of iterations.

5.2 Anisotropic diffusion-reaction problem

We discuss the results obtained on the anisotropic diffusion-reaction problem (5) using $\varepsilon = 10^{-i}$, $i = 0, 2, 4$. Our purpose is to evaluate the behaviour of our preconditioners as the difference among the differential operators, due to the different anisotropies, increases. In this case we chose as a primary matrix the one arising from the discretization of the negative Laplacian, i.e., the matrix corresponding to the isotropic differential operator in the PDE system (5). This makes the multilevel transfer operators independent of ε, as for problem 1. We set $n = 1000$.

In Table 4 we report the number of iterations of the CG solver preconditioned with BJAC and SMMLS. As expected, with the SMMLS preconditioner the number of CG iterations generally increases as ε decreases. However, this number is always much smaller than with BJAC, yielding a smaller execution time for all the values of ε, as shown in Figure 3 (left). For $\varepsilon = 1, 10^{-2}$ the number of iterations is almost constant as the number of processors varies; conversely, for $\varepsilon = 10^{-4}$ a growth of the number of iterations is already visible on 8 processors. We note that the significant reduction of the number of iterations observed on one processor for $\varepsilon = 10^{-4}$ is a consequence of the greater effectiveness of the basic Schwarz preconditioner within our multilevel framework (the block-Jacobi method reduces to an incomplete factorization when one processor is considered). The speedups corresponding to the anisotropic problem with $\varepsilon = 10^{-2}$ are plotted in Figure 3 (right). We see that a satisfactory speedup is obtained with both the 4LEV and 5LEV preconditioners.

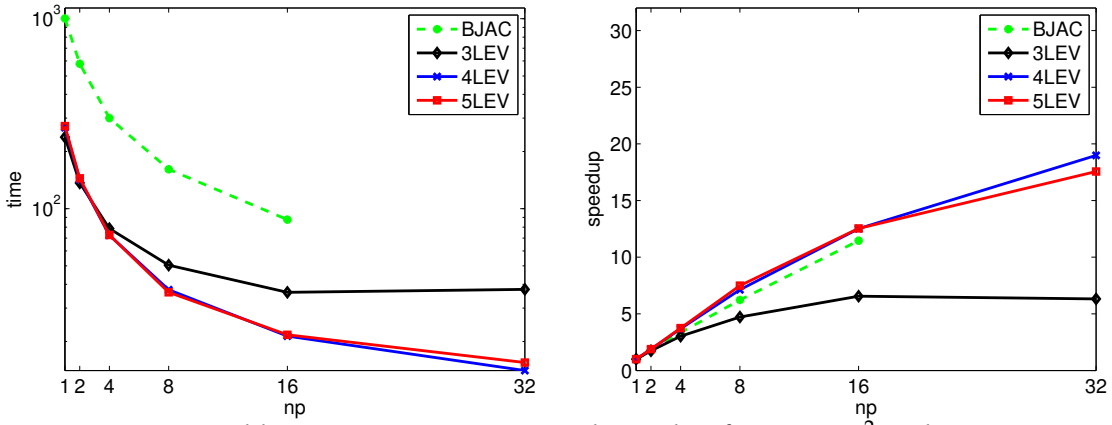

Figure 3: Problem 2 - execution time and speedup for $\varepsilon = 10^{-2}$ and $n = 1000$.

ε	3LEV	4LEV	5LEV
Classical			
1	8	8	8
10^{-2}	18	19	22
10^{-4}	21	22	21
New			
1	7	9	10
10^{-2}	29	32	32
10^{-4}	12	12	12

Table 5: Problem 2 - number of iterations obtained on one processor with the classical block version (Classical) and our version (New) of SMMLS, for $n = 500$ and $\vartheta = 10^{-2}$.

We conclude this section comparing our preconditioners with the Trilinos/ML preconditioners using the classical block approach. The experiments were performed on one processor, with $n = 500$ and $\vartheta = 0, 10^{-1}, 10^{-2}, 10^{-3}$. In Table 5 we report the number of iterations obtained with $\vartheta = 10^{-2}$, which generally produced the better results for almost all the values of ε. We see that for $\varepsilon = 10^{-2}$ the classical block approach is more effective; nevertheless, the number of iterations obtained with our preconditioners is fairly good, especially if we take into account the lower computational cost that results from building the prolongation and restriction operators by using a primary matrix associated with a single PDE. As previously noted, for $\varepsilon = 10^{-4}$ the results obtained by our preconditioner on a single processor are not representative of its general behaviour.

6 Convergence analysis

In this section, we discuss the convergence properties of the AMLS preconditioners presented in Section 3. Specifically, we show how to extend classical convergence results of additive multilevel Schwarz preconditioners to our schemes (see, e.g., [8, 27, 33]).

For each level l, let $\mathcal{W}_l$ be the vector space of the grid functions $\mathbf{u}_l$ associated with the index space $\mathcal{V}_l$, and let $\mathcal{W}_{l,j}$, $j = 1, \ldots, t_l$, be the vector spaces of the grid functions $\mathbf{u}_{l,j}$ associated with the index sets $\mathcal{V}_{l,j}$ of the overlapping partition of $\mathcal{V}_l$ used by the AS method. We denote with

$(\cdot,\cdot)_l$ the discrete L^2 scalar product either in $\mathcal{W}_l$ or $\mathcal{W}_{l,j}$ and by $\|\cdot\|_l = (\cdot,\cdot)_l^{1/2}$ the corresponding discrete L^2-norms; furthermore, we denote with $\|\cdot\|_{\hat{\mathcal{A}}}$ the energy norm associated with the matrix $\hat{\mathcal{A}}_l$ in (8). Analogously, we denote with W_l the vector space of the grid functions u_l associated with the index set V_l corresponding to the primary matrix A_l at level l, and with $W_{l,j}$, $j = 1,\dots,t_l$, the vector spaces associated with the index sets $V_{l,j}$, all endowed with the corresponding L^2 scalar products, denoted again by $(\cdot,\cdot)_l$. We note that $\mathcal{W}_l$ can be regarded as the Cartesian product of n_e spaces $W_l^{(i)}$, each isomorphic to W_l; analogously, $\mathcal{W}_{l,j}$ can be regarded as the Cartesian product of n_e spaces $W_{l,j}^{(i)}$ isomorphic to $W_{l,j}$. Therefore, we can identify each space $W_l^{(i)}$ with W_l and each space $W_{l,j}^{(i)}$ with $W_{l,j}$. This allows to express the L^2 scalar product of $\mathbf{u}_l, \mathbf{v}_l \in \mathcal{W}_l$ in terms of the scalar products of their components $u_l^{(i)}, w_l^{(i)}$, $i = 1,\dots,n_e$, as follows

$$(\mathbf{u}_l, \mathbf{v}_l)_l = \left(u_l^{(i)}, w_l^{(i)}\right)_l + \cdots + \left(u_l^{(n_e)}, w_l^{(n_e)}\right)_l.$$

Similarly, the L^2 scalar product in $\mathcal{W}_{l,j}$ can be written in terms of the scalar product in $W_{l,j}$.

The multilevel restriction and prolongation operators defined in section 3 are such that $\hat{\mathcal{R}}_l : \mathcal{W}_l \to \mathcal{W}_{l-1}$, $\hat{\mathcal{P}}_l : \mathcal{W}_{l-1} \to \mathcal{W}_l$ and

$$(\hat{\mathcal{R}}_l \mathbf{u}_l, \mathbf{v}_{l-1})_{l-1} = (\mathbf{u}_l, \hat{\mathcal{P}}_l \mathbf{v}_{l-1})_l, \tag{18}$$

for all $\mathbf{u}_l \in \mathcal{W}_l$ and $\mathbf{v}_{l-1} \in \mathcal{W}_{l-1}$. Similar results hold for the restriction and prolongation operators $\hat{\mathcal{R}}_{l,j}$ and $\hat{\mathcal{P}}_{l,j}$ of the AS method at each level l. Furthermore, we require that the operator $\mathcal{S}_{l,j} : \mathcal{W}_{l,j} \to \mathcal{W}_{l,j}$ that approximates the inverse of $\mathcal{A}_{l,j}$ be symmetric positive definite for all j, although $\mathcal{A}_{l,j}$ may not have this property. Indeed, while this assumption appears restrictive, it greatly facilitates the analysis. On the other hand, it is possible to construct smoothers that are symmetric and the practice shows agreement with our theoretical statements also when using nonsymmetric smoothers.

By substituting $\mathcal{A}_{l,j}^{-1}$ with $\mathcal{S}_{l,j}$, the AMLS preconditioner in (16) takes the following form

$$\mathcal{M} = \sum_{l=0}^{L} \sum_{j=1}^{t_l} \mathcal{P}_{l,j} \mathcal{S}_{l,j} \mathcal{R}_{l,j},$$

where we neglect the superscript $AMLS$ in $\mathcal{M}$ for brevity. Thus, letting

$$\mathcal{E}_{l,j} = \mathcal{S}_{l,j} \mathcal{R}_{l,j} \mathcal{A}, \tag{19}$$

we get

$$\mathcal{M}\mathcal{A} = \sum_{l=0}^{L} \sum_{j=1}^{t_l} \mathcal{P}_{l,j} \mathcal{E}_{l,j}.$$

For simplicity, we consider the twolevel preconditioner. In this case it follows from (15) that $\mathcal{P}_{l,j} = \hat{\mathcal{P}}_{l,j}$ and $\mathcal{R}_{l,j} = \hat{\mathcal{R}}_{l,j}$; furthermore, we can conveniently omit the index l and identify the coarse level with $j = 0$. Therefore, in the twolevel case $\mathcal{M}$ and $\mathcal{M}\mathcal{A}$ can be written as

$$\mathcal{M} = \sum_{j=0}^{t} \hat{\mathcal{P}}_j \mathcal{S}_j \hat{\mathcal{R}}_j, \quad \mathcal{M}\mathcal{A} = \sum_{j=0}^{t} \hat{\mathcal{P}}_j \mathcal{E}_j. \tag{20}$$

We also recall that $\mathcal{A} = \hat{\mathcal{A}} + \mathcal{B}$ and suppose that $\hat{\mathcal{A}}$ is symmetric positive definite.

We first discuss the convergence properties of the preconditioner $\mathcal{M}$ in the case where $\mathcal{A}$ is symmetric positive definite. We follow the same line of reasoning as [8], but we make slightly different assumptions that take into account the role of $\hat{\mathcal{A}}$ in the costruction of $\mathcal{M}$. These assumptions are specified next:

(A.1) *there exists a constant* $\xi > 0$ *such that for all* $\mathbf{u} \in \mathcal{W}$ *we can find* $\mathbf{u}_j \in \mathcal{W}_j$, $j = 0, \ldots, t$, *satisfying* $\mathbf{u} = \sum_{j=0}^{t} \hat{\mathcal{P}}_j \mathbf{u}_j$ *and*

$$\sum_{j=0}^{t} \left(\mathcal{S}_j^{-1} \mathbf{u}_j, \mathbf{u}_j \right) < \xi \ (\hat{\mathcal{A}}\, \mathbf{u}, \mathbf{u}) \,;$$

(A.2) *there exists a constant* $\omega > 0$ *such that for all* $\mathbf{u}_j \in \mathcal{W}_j$, $j = 0, \ldots, t$, *we have*

$$(\hat{\mathcal{A}}\, \hat{\mathcal{P}}_j \mathbf{u}_j, \hat{\mathcal{P}}_j \mathbf{u}_j) \leq \omega \left(\mathcal{S}_j^{-1} \mathbf{u}_j, \mathbf{u}_j \right) ;$$

(A.3) *there exists a constant* $\alpha_0 > 0$ *such that, for all* $\mathbf{u} \in \mathcal{W}$ *and* $\mathbf{u}_j \in \mathcal{W}_j$, $j = 1, \ldots, t$, *we have*

$$\sum_{j=1}^{t} (\hat{\mathcal{A}}\mathbf{u}, \hat{\mathcal{P}}_j \mathbf{u}_j) \leq \alpha_0^{1/2} \, (\hat{\mathcal{A}}\mathbf{u}, \mathbf{u})^{1/2} \left(\sum_{j=1}^{t} (\hat{\mathcal{A}} \hat{\mathcal{P}}_j \mathbf{u}_j, \hat{\mathcal{P}}_j \mathbf{u}_j) \right)^{1/2} ;$$

(A.4) *for any operator* $\mathcal{H}$ *such that* $(\hat{\mathcal{A}}\mathbf{u}, \mathcal{H}\mathbf{u}) > 0$ *for all nonzero* $\mathbf{u} \in \mathcal{W}$, *there exists a constant* $\mu \in (0, 1)$ *such that for all* $\mathbf{u} \in \mathcal{W}$, *we have*

$$|(\mathcal{B}\mathbf{u}, \mathcal{H}\mathbf{u})| \leq \mu \, (\hat{\mathcal{A}}\mathbf{u}, \mathcal{H}\mathbf{u}); \tag{21}$$

furthermore, there exists a constant $\gamma > 0$ *such that for all* $\mathbf{u}, \mathbf{v} \in \mathcal{W}$, *we have*

$$|(\mathcal{B}\mathbf{u}, \mathbf{v})| \leq \gamma (\hat{\mathcal{A}}\mathbf{u}, \mathbf{u})^{1/2} (\hat{\mathcal{A}}\mathbf{v}, \mathbf{v})^{1/2}.$$

All the constants $\xi, \omega, \alpha_0, \mu$ and γ are assumed to be independent of t. Furthermore, without loss of generality, we consider $\xi, \omega, \alpha_0 \geq 1$.

Assumption (A.1) means that any function in the solution space $\mathcal{W}$ can be decomposed into a sum of functions in $\mathcal{W}_j$ and this decomposition is stable with respect to the energy norm induced by $\hat{\mathcal{A}}$. Assumption (A.4) means that $\mathcal{B}$ can be regarded as a small perturbation of $\hat{\mathcal{A}}$; in particular, inequality (21) implies that

$$(1 - \mu)(\hat{\mathcal{A}}\mathbf{u}, \mathcal{H}\mathbf{u}) \leq (\mathcal{A}\mathbf{u}, \mathcal{H}\mathbf{u}) \leq (1 + \mu)(\hat{\mathcal{A}}\mathbf{u}, \mathcal{H}\mathbf{u}). \tag{22}$$

Assumptions (A.2) and (A.4) ensure that the largest eigenvalue of $\mathcal{S}_j \mathcal{A}_j$ is bounded by ω, i.e., $\mathcal{S}_j$ does not provide a 'too bad' approximation to $\mathcal{A}_j^{-1}$. Finally, (A.3) requires that the overlapping of the spaces $\mathcal{W}_j$ is bounded in terms of the energy norm induced by $\hat{\mathcal{A}}$.

The following theorem provides a bound on the condition number of $\mathcal{MA}$.

Theorem 1 *Let* $\mathcal{A}$ *be symmetric positive definite. Under the assumptions (A.1)-(A.4), with* $\mathcal{MA}$ *satisfying the assumptions for* $\mathcal{H}$ *in (A.4), it holds:*

$$\kappa(\mathcal{MA}) \leq \frac{\omega \xi (1 + \alpha_0^{1/2})^2 (1 + \mu)^3}{(1 - \mu)^3},$$

where $\kappa(\mathcal{MA})$ *is the spectral condition number of* $\mathcal{MA}$.

Proof. Following [8], we first estimate the smallest eigenvalue of $\mathcal{MA}$. By using assumptions (A.1) and (A.4), as well as (18), (19) and (20), we get

$$\begin{aligned}
(\hat{\mathcal{A}}\mathbf{u},\mathbf{u}) &= (\hat{\mathcal{A}}\mathbf{u},\sum_{j=0}^{t}\hat{\mathcal{P}}_j\mathbf{u}_j) = (\mathcal{A}\mathbf{u},\sum_{j=0}^{t}\hat{\mathcal{P}}_j\mathbf{u}_j) - (\mathcal{B}\mathbf{u},\sum_{j=0}^{t}\hat{\mathcal{P}}_j\mathbf{u}_j)\\
&= \sum_{j=0}^{t}(\hat{\mathcal{R}}_j\mathcal{A}\mathbf{u},\mathbf{u}_j) - \sum_{j=0}^{t}(\hat{\mathcal{R}}_j\mathcal{B}\mathbf{u},\mathbf{u}_j)\\
&= \sum_{j=0}^{t}(\mathcal{S}_j^{-1}\mathcal{E}_j\mathbf{u},\mathbf{u}_j) - \sum_{j=0}^{t}(\hat{\mathcal{R}}_j\mathcal{B}\mathbf{u},\mathbf{u}_j)\\
&\le \left(\sum_{j=0}^{t}(\mathcal{S}_j^{-1}\mathbf{u}_j,\mathbf{u}_j)\right)^{1/2}\left(\sum_{j=0}^{t}(\mathcal{S}_j^{-1}\mathcal{E}_j\mathbf{u},\mathcal{E}_j\mathbf{u})\right)^{1/2} - (\mathcal{B}\mathbf{u},\mathbf{u})\\
&\le \xi^{1/2}(\hat{\mathcal{A}}\mathbf{u},\mathbf{u})^{1/2}\left(\sum_{j=0}^{t}(\hat{\mathcal{R}}_j\mathcal{A}\mathbf{u},\mathcal{E}_j\mathbf{u})\right)^{1/2} + \mu(\hat{\mathcal{A}}\mathbf{u},\mathbf{u})\\
&= \xi^{1/2}(\hat{\mathcal{A}}\mathbf{u},\mathbf{u})^{1/2}(\mathcal{A}\mathbf{u},\mathcal{MA}\mathbf{u})^{1/2} + \mu(\hat{\mathcal{A}}\mathbf{u},\mathbf{u}).
\end{aligned}$$

It follows that

$$(1-\mu)^2(\hat{\mathcal{A}}\mathbf{u},\mathbf{u}) \le \xi(\mathcal{A}\mathbf{u},\mathcal{MA}\mathbf{u}) \tag{23}$$

and thus, by (22), we have

$$(\mathcal{A}\mathbf{u},\mathbf{u}) \le \frac{\xi(1+\mu)}{(1-\mu)^2}\,(\mathcal{A}\mathbf{u},\mathcal{MA}\mathbf{u}), \tag{24}$$

which gives the following estimate of the smallest eigenvalue of the preconditioned matrix:

$$\lambda_{min}(\mathcal{MA}) \ge \frac{(1-\mu)^2}{\xi(1+\mu)}. \tag{25}$$

Next, we provide an upper bound on the largest eigenvalue of $\mathcal{MA}$. We first observe that, by assumption (A.3), we have

$$\begin{aligned}
&\sum_{j=0}^{t}(\hat{\mathcal{A}}\mathbf{u},\hat{\mathcal{P}}_j\mathbf{u}_j) = (\hat{\mathcal{A}}\mathbf{u},\hat{\mathcal{P}}_0\mathbf{u}_0) + \sum_{j=1}^{t}(\hat{\mathcal{A}}\mathbf{u},\hat{\mathcal{P}}_j\mathbf{u}_j)\\
\le\ & (\hat{\mathcal{A}}\mathbf{u},\mathbf{u})^{1/2}(\hat{\mathcal{A}}\hat{\mathcal{P}}_0\mathbf{u}_0,\hat{\mathcal{P}}_0\mathbf{u}_0)^{1/2} + \alpha_0^{1/2}(\hat{\mathcal{A}}\mathbf{u},\mathbf{u})^{1/2}\left(\sum_{j=1}^{t}(\hat{\mathcal{A}}\hat{\mathcal{P}}_j\mathbf{u}_j,\hat{\mathcal{P}}_j\mathbf{u}_j)\right)^{1/2}\\
\le\ & (1+\alpha_0^{1/2})\,(\hat{\mathcal{A}}\mathbf{u},\mathbf{u})^{1/2}\left(\sum_{j=0}^{t}(\hat{\mathcal{A}}\hat{\mathcal{P}}_j\mathbf{u}_j,\hat{\mathcal{P}}_j\mathbf{u}_j)\right)^{1/2}.
\end{aligned}$$

Then, by exploiting assumptions (A.2) and (A.4), and equalities (18), (19) and (20), we get

$$\begin{aligned}
(\hat{\mathcal{A}}\mathbf{u}, \mathcal{M}\mathcal{A}\mathbf{u}) &= \sum_{j=0}^{t} (\hat{\mathcal{A}}\mathbf{u}, \hat{\mathcal{P}}_j \mathcal{E}_j \mathbf{u}) \\
&\leq (1+\alpha_0^{1/2})\,(\hat{\mathcal{A}}\mathbf{u}, \mathbf{u})^{1/2} \left(\sum_{j=0}^{t} (\hat{\mathcal{A}}\hat{\mathcal{P}}_j \mathcal{E}_j \mathbf{u}, \hat{\mathcal{P}}_j \mathcal{E}_j \mathbf{u}) \right)^{1/2} \\
&\leq \omega^{1/2}(1+\alpha_0^{1/2})\,(\hat{\mathcal{A}}\mathbf{u}, \mathbf{u})^{1/2} \left(\sum_{j=0}^{t} (\mathcal{S}_j^{-1} \mathcal{E}_j \mathbf{u}, \mathcal{E}_j \mathbf{u}) \right)^{1/2} \\
&= \omega^{1/2}(1+\alpha_0^{1/2})\,(\hat{\mathcal{A}}\mathbf{u}, \mathbf{u})^{1/2} \left(\sum_{j=0}^{t} (\mathcal{A}\mathbf{u}, \hat{\mathcal{P}}_j \mathcal{E}_j \mathbf{u}) \right)^{1/2} \\
&= \omega^{1/2}(1+\alpha_0^{1/2})\,(\hat{\mathcal{A}}\mathbf{u}, \mathbf{u})^{1/2} (\mathcal{A}\mathbf{u}, \mathcal{M}\mathcal{A}\mathbf{u})^{1/2}.
\end{aligned}$$

By (22) it follows that

$$(\mathcal{A}\mathbf{u}, \mathcal{M}\mathcal{A}\mathbf{u}) \leq \frac{\omega(1+\alpha_0^{1/2})^2(1+\mu)^2}{1-\mu} (\mathcal{A}\mathbf{u}, \mathbf{u})$$

and hence

$$\lambda_{max}(\mathcal{M}\mathcal{A}) \leq \frac{\omega(1+\alpha_0^{1/2})^2(1+\mu)^2}{1-\mu}.$$

The previous inequality, together with (25), yields the thesis. □

In the general case of $\mathcal{A}$ being nonsymmetric, one can apply a nonsymmetric Krylov solver, such as GMRES or BiCGStab, to solve the preconditioned system

$$\mathcal{M}\mathcal{A}\mathbf{u} = \mathcal{M}\mathbf{f}.$$

In particular, for the GMRES method we have the following convergence estimate [14]:

$$\|\mathbf{r}^m\|_{\hat{\mathcal{A}}}^2 \leq \left(1 - \frac{\beta_1^2}{\beta_2^2}\right)^m \|\mathbf{r}^0\|_{\hat{\mathcal{A}}}^2,$$

where $\mathbf{r}^m = \mathcal{M}\mathbf{f} - \mathcal{M}\mathcal{A}\mathbf{u}^m$ is the residual corresponding to the mth iterate, and β_1 and β_2 are defined as follows:

$$\beta_1 = \min_{\mathbf{u}\neq 0} \frac{(\hat{\mathcal{A}}\mathbf{u}, \mathcal{M}\mathcal{A}\mathbf{u})}{\|\mathbf{u}\|_{\hat{\mathcal{A}}}^2}, \quad \beta_2 = \max_{\mathbf{u}\neq 0} \frac{\|\mathcal{M}\mathcal{A}\mathbf{u}\|_{\hat{\mathcal{A}}}}{\|\mathbf{u}\|_{\hat{\mathcal{A}}}}.$$

Notice that β_1 is required to be positive to get convergence of the preconditioned GMRES method [14]. Furthermore, from the Cauchy-Schwarz inequality we have $\beta_1/\beta_2 \leq 1$.

To provide a lower bound on β_1/β_2, we make a further assumption:

(A.5) *there exists a constant* $\alpha_1 > 0$ *such that, for any* $\mathbf{u} \in \mathcal{W}$ *and* $\mathbf{u}_j \in \mathcal{W}_j$, $j = 1, \ldots, t$, *it is*

$$\sum_{j=1}^{t} (\mathcal{B}\mathbf{u}, \hat{\mathcal{P}}_j \mathbf{u}_j) \leq \alpha_1^{1/2} (\hat{\mathcal{A}}\mathbf{u}, \mathbf{u})^{1/2} \left(\sum_{j=1}^{t} (\hat{\mathcal{A}}\hat{\mathcal{P}}_j \mathbf{u}_j, \hat{\mathcal{P}}_j \mathbf{u}_j) \right)^{1/2}. \tag{26}$$

This means that $\mathcal{B}$ is 'bounded' by $\hat{\mathcal{A}}$ in some sense. As the costants in (A.1)-(A.4), α_1 is assumed to be independent of t. The following theorem holds:

Theorem 2 *Under the assumptions (A.1)-(A.5), with $\mathcal{MA}$ satisfying the assumptions for $\mathcal{H}$ in (A.4),*

$$\frac{\beta_1}{\beta_2} \geq \frac{(1-\mu)^2}{2\sqrt{2}\,\omega\,\xi\,\alpha_0^{1/2}(1+\mu)(1+\gamma+\alpha_0^{1/2}+\alpha_1^{1/2})} > 0.$$

Proof. From estimate (23), which holds because by hypothesis $(\hat{\mathcal{A}}\mathbf{u}, \mathcal{MA}\mathbf{u}) > 0$ for $\mathbf{u} \neq 0$, and from (22) it follows that

$$\beta_1 \geq \frac{(1-\mu)^2}{\xi(1+\mu)}. \tag{27}$$

Next, we estimate β_2 following the procedure illustrated in [8]. Consider any $\mathbf{u} \in \mathcal{W}$ and let $\mathbf{w} = \sum_{j=1}^{t} \hat{\mathcal{P}}_j \mathcal{E}_j \mathbf{u}$. By using assumption (A.3), we have

$$\|\mathbf{w}\|_{\hat{\mathcal{A}}}^2 = (\hat{\mathcal{A}}\mathbf{w}, \sum_{j=1}^{t} \hat{\mathcal{P}}_j \mathcal{E}_j \mathbf{u}) = \sum_{j=1}^{t} (\hat{\mathcal{A}}\mathbf{w}, \hat{\mathcal{P}}_j \mathcal{E}_j \mathbf{u}) \leq \alpha_0^{1/2} \|\mathbf{w}\|_{\hat{\mathcal{A}}} \left(\sum_{j=1}^{t} \|\hat{\mathcal{P}}_j \mathcal{E}_j \mathbf{u}\|_{\hat{\mathcal{A}}}^2 \right)^{1/2}.$$

It follows that

$$\|\mathbf{w}\|_{\hat{\mathcal{A}}} \leq \alpha_0^{1/2} \left(\sum_{j=1}^{t} \|\hat{\mathcal{P}}_j \mathcal{E}_j \mathbf{u}\|_{\hat{\mathcal{A}}}^2 \right)^{1/2}.$$

By using the previous inequality, the expression of $\mathcal{MA}$ in (20) and assumption (A.2), we get

$$\begin{aligned} \|\mathcal{MA}\mathbf{u}\|_{\hat{\mathcal{A}}}^2 &= \|\sum_{j=0}^{t} \hat{\mathcal{P}}_j \mathcal{E}_j \mathbf{u}\|_{\hat{\mathcal{A}}}^2 \leq 2\|\hat{\mathcal{P}}_0 \mathcal{E}_0 \mathbf{u}\|_{\hat{\mathcal{A}}}^2 + 2\|\sum_{j=1}^{t} \hat{\mathcal{P}}_j \mathcal{E}_j \mathbf{u}\|_{\hat{\mathcal{A}}}^2 \\ &\leq 2\|\hat{\mathcal{P}}_0 \mathcal{E}_0 \mathbf{u}\|_{\hat{\mathcal{A}}}^2 + 2\alpha_0 \sum_{j=1}^{t} \|\hat{\mathcal{P}}_j \mathcal{E}_j \mathbf{u}\|_{\hat{\mathcal{A}}}^2 \leq 2\omega\alpha_0 \sum_{j=0}^{t} (\mathcal{S}_j^{-1} \mathcal{E}_j \mathbf{u}, \mathcal{E}_j \mathbf{u}). \end{aligned} \tag{28}$$

By using assumptions (A.2), (A.3) and (A.4), as well as equalities (18) and (19), and inequal-

ity (26), we obtain

$$
\begin{aligned}
&\sum_{j=0}^{t}(\mathcal{S}_j^{-1}\mathcal{E}_j\mathbf{u},\mathcal{E}_j\mathbf{u}) = \sum_{j=0}^{t}(\hat{\mathcal{R}}_j\mathcal{A}\mathbf{u},\mathcal{E}_j\mathbf{u}) = \sum_{j=0}^{t}(\mathcal{A}\mathbf{u},\hat{\mathcal{P}}_j\mathcal{E}_j\mathbf{u}) \\
= &\sum_{j=0}^{t}(\hat{\mathcal{A}}\mathbf{u},\hat{\mathcal{P}}_j\mathcal{E}_j\mathbf{u}) + \sum_{j=0}^{t}(\mathcal{B}\mathbf{u},\hat{\mathcal{P}}_j\mathcal{E}_j\mathbf{u}) \\
= &(\hat{\mathcal{A}}\mathbf{u},\hat{\mathcal{P}}_0\mathcal{E}_0\mathbf{u}) + \sum_{j=1}^{t}(\hat{\mathcal{A}}\mathbf{u},\hat{\mathcal{P}}_j\mathcal{E}_j\mathbf{u}) + (\mathcal{B}\mathbf{u},\hat{\mathcal{P}}_0\mathcal{E}_0\mathbf{u}) + \sum_{j=1}^{t}(\mathcal{B}\mathbf{u},\hat{\mathcal{P}}_j\mathcal{E}_j\mathbf{u}) \\
\leq &(1+\gamma)\|\mathbf{u}\|_{\hat{\mathcal{A}}}\|\hat{\mathcal{P}}_0\mathcal{E}_0\mathbf{u}\|_{\hat{\mathcal{A}}} + (\alpha_0^{1/2}+\alpha_1^{1/2})\|\mathbf{u}\|_{\hat{\mathcal{A}}}\left(\sum_{j=1}^{t}\|\hat{\mathcal{P}}_j\mathcal{E}_j\mathbf{u}\|_{\hat{\mathcal{A}}}^2\right)^{1/2} \\
\leq &2\,(1+\gamma+\alpha_0^{1/2}+\alpha_1^{1/2})\,\|\mathbf{u}\|_{\hat{\mathcal{A}}}\left(\sum_{j=0}^{t}\|\hat{\mathcal{P}}_j\mathcal{E}_j\mathbf{u}\|_{\hat{\mathcal{A}}}^2\right)^{1/2} \\
\leq &2\omega^{1/2}\,(1+\gamma+\alpha_0^{1/2}+\alpha_1^{1/2})\,\|\mathbf{u}\|_{\hat{\mathcal{A}}}\left(\sum_{j=0}^{t}(\mathcal{S}_j^{-1}\mathcal{E}_j\mathbf{u},\mathcal{E}_j\mathbf{u})\right)^{1/2}.
\end{aligned}
$$

Therefore, we get the following bound:

$$
\sum_{j=0}^{t}(\mathcal{S}_j^{-1}\mathcal{E}_j\mathbf{u},\mathcal{E}_j\mathbf{u}) \leq 4\omega\,(1+\gamma+\alpha_0^{1/2}+\alpha_1^{1/2})^2\,\|\mathbf{u}\|_{\hat{\mathcal{A}}}^2
$$

By using (28), we have

$$
\|\mathcal{M}\mathcal{A}\mathbf{u}\|_{\hat{\mathcal{A}}}^2 \leq 8\omega^2\alpha_0\,(1+\gamma+\alpha_0^{1/2}+\alpha_1^{1/2})^2\,\|\mathbf{u}\|_{\hat{\mathcal{A}}}^2
$$

and hence

$$
\beta_2 \leq 2\sqrt{2}\omega\alpha_0^{1/2}(1+\gamma+\alpha_0^{1/2}+\alpha_1^{1/2}).
$$

The thesis follows from the previous bound and (27). □

Clearly, in specific applications we must verify that assumptions (A.1)-(A.5) hold. To this aim, the block-diagonal form of $\hat{\mathcal{A}}$ and $\hat{\mathcal{P}}$ (see (8), (9) and (13)) and the fact that the scalar products in $\mathcal{W}$ and in $\mathcal{W}_j$, $j = 1,\ldots,t$, can be expressed in terms of the scalar products in the corresponding spaces associated with the primary matrix (i.e., W and W_j) may be of great help. In particular, we can exploit results concerning AMLS methods for scalar elliptic PDE problems (see, e.g., [4, 8]).

7 Conclusions

A point-based approach for building algebraic multilevel Schwarz preconditioners based on smoothed aggregation was discussed. These preconditioners were designed for solving algebraic problems arising from the discretization of a class of systems of elliptic PDEs with zero-order coupling terms. In this approach, the restriction and prolongation operators associated with the multilevel hierarchy and with the Schwarz matrix partitioning were defined based on a

primary matrix corresponding to the second-order differential operator of a single PDE. Further, a parallel implementation of the proposed preconditioners was illustrated.

Numerical experiments with elliptic optimality systems and with multi-species anisotropic diffusion-reaction models showed the effectiveness and the parallel efficiency of the proposed preconditioners. Convergence results for the twolevel version of the proposed additive Schwarz preconditioner were presented.

References

[1] P. Bastian, W. Hackbusch, and G. Wittum, *Additive and Multiplicative Multigrid – A Comparison*, Computing, 60 (1998), pp. 345–364.

[2] A. Borzì and G. Borzì, *An algebraic multigrid method for a class of elliptic differential systems*, SIAM J. Sci. Comput., 25 (2003), pp. 302–323.

[3] A. Brandt, *Multigrid techniques: 1984 guide with applications to fluid dynamics*, vol. 85 of GMD-Studien, Gesellschaft für Mathematik und Datenverarbeitung mbH, St. Augustin, 1984.

[4] M. Brezina and P. Vaněk, *A black-box iterative solver based on a two-level Schwarz method*, Computing, 63 (1999), pp. 233–263.

[5] M. Brezina, P. Vaněk, and P. S. Vassilevski, *An improved convergence analysis of smoothed aggregation algebraic multigrid*, Numer. Linear Algebra Appl., 19 (2012), pp. 441–469.

[6] A. Buttari, P. D'Ambra, D. di Serafino, and S. Filippone, *2LEV-D2P4: a package of high-performance preconditioners for scientific and engineering applications*, Appl. Algebra Engrg. Comm. Comput., 18 (2007), pp. 223–239.

[7] X.-C. Cai and Y. Saad, *Overlapping domain decomposition algorithms for general sparse matrices*, Numer. Linear Algebra Appl., 3 (1996), pp. 221–237.

[8] T. F. Chan and J. Zou, *A convergence theory of multilevel additive Schwarz methods on unstructured meshes*, Numer. Algorithms, 13 (1996), pp. 365–398.

[9] P. D'Ambra, D. di Serafino, and S. Filippone, *On the development of PSBLAS-based parallel two-level Schwarz preconditioners*, Appl. Numer. Math., 57 (2007), pp. 1181–1196.

[10] P. D'Ambra, D. di Serafino, and S. Filippone, *Performance analysis of parallel Schwarz preconditioners in the LES of turbulent channel flows*, Comput. Math. Appl., 65 (2013), pp. 352–361.

[11] P. D'Ambra, D. di Serafino, and S. Filippone, *MLD2P4: a package of parallel algebraic multilevel domain decomposition preconditioners in Fortran 95*, ACM Trans. Math. Software, 37 (2010), art. 30, 23 pages.

[12] T. A. Davis, *Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method*, ACM Trans. Math. Software, 30 (2004), pp. 196–199.

[13] J. Dongarra and R. C. Whaley, *A user's guide to the BLACS v1.1, Lapack Working Note 94*, Tech. Report UT-CS-95-281, University of Tennessee, March 1995 (updated May 1997).

[14] S. Eisenstat, H. Elman, and M. Schultz, *Variational iterative methods for nonsymmetric systems of linear equations*, SIAM J. Numer. Anal., 20 (1983), pp. 345–357.

[15] R. D. Falgout, *An introduction to algebraic multigrid*, Computing in Science and Engineering, 8 (2006), pp. 24–33.

[16] S. Filippone and A. Buttari, *PSBLAS: User's and Reference Guide*, Available from http://www.ce.uniroma2.it/psblas/, 2008.

[17] S. Filippone and M. Colajanni, *PSBLAS: a library for parallel linear algebra computation on sparse matrices*, ACM Trans. Math. Software, 26 (2000), pp. 527–550.

[18] T. Füllenbach and K. Stüben, *Algebraic multigrid for selected PDE systems*, in Elliptic and parabolic problems (Rolduc/Gaeta, 2001), World Sci. Publ., River Edge, NJ, 2002, pp. 399–410.

[19] M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, and M. G. Sala, *ML 5.0 smoothed aggregation user's guide*, Tech. Report SAND2006-2649, Sandia National Laboratories, Albuquerque, NM, and Livermore, CA, USA, 2006.

[20] M. Griebel, D. Oeltz, and M. A. Schweitzer, *An algebraic multigrid method for linear elasticity*, SIAM J. Sci. Comput., 25 (2003), pp. 385–407.

[21] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley, *An overview of the Trilinos Project*, ACM Trans. Math. Soft., 31 (2005), pp. 397–423.

[22] D. Lahaye, H. De Gersen, S. Vandewalle, and K. Hameyer, *Algebraic multigrid for complex symmetric systems*, IEEE Trans. on Magnetics, 36 (2000), pp. 1535–1538.

[23] J.-L. Lions, *Optimal control of systems governed by partial differential equations.*, Springer-Verlag, New York, 1971.

[24] J. Mandel, M. Brezina, and P. Vaněk, *Energy optimization of algebraic multigrid bases*, Computing, 62 (1999), pp. 205–228.

[25] F. Rothe, *Global Solutions of Reaction-Diffusion Systems*, vol. 1072 of Lecture Notes in Mathematics, Springer-Verlag, Berlin, 1984.

[26] J. Ruge and K. Stüben, *Algebraic multigrid (AMG)*, in Multigrid methods, S. F. McCormick, ed., vol. 3 of Frontiers in Applied Mathematics, SIAM, 1987.

[27] B. F. Smith, P. E. Bjørstad, and W. D. Gropp, *Domain Decomposition*, Cambridge University Press, Cambridge, 1996.

[28] M. Snir, S. Otto, S. Huss-Lederman, D. W. Walker, and J. J. Dongarra, *MPI: The Complete Reference. Vol. 1 – The MPI Core*, Scientific and Engineering Computation, The MIT Press, Cambridge, MA, second ed., 1998.

[29] K. Stüben, *An introduction to algebraic multigrid*, in Multigrid, U. Trottenberg, C. Oosterlee, and A. Schüller, eds., Academic Press, 2001, pp. 413–532.

[30] A. Toselli and O. Widlund, *Domain Decomposition Methods - Algorithms and Theory*, Springer, 2005.

[31] R. S. Tuminaro and C. Tong, *Parallel smoothed aggregation multigrid: Aggregation strategies on massively parallel machines*, in Proceedings of the 2000 ACM/IEEE conference on Supercomputing, Dallas, TX, USA, 2000, CDROM.

[32] P. Vaněk, J. Mandel, and M. Brezina, *Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems*, Computing, 56 (1996), pp. 179–196.

[33] J. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM Rev., 34 (1992), pp. 581–613.

[34] I. Yavneh, *A method for devising efficient multigrid smoothers for complicated PDE systems*, SIAM J. Sci. Comput., 14 (1993), pp. 1437–1463.

[35] I. Yavneh, *Coarse-grid correction for nonelliptic and singular perturbation problems*, SIAM J. Sci. Comput., 19 (1998), pp. 1682–1699.