Condition number estimates and weak scaling for 2-level 2-lagrange multiplier methods for general domains and cross points

SIAM J. SCI. COMPUT.
Vol. 37, No. 2, pp. C247–C267

CONDITION NUMBER ESTIMATES AND WEAK SCALING FOR 2-LEVEL 2-LAGRANGE MULTIPLIER METHODS FOR GENERAL DOMAINS AND CROSS POINTS*

ANASTASIOS KARANGELIS† AND SÉBASTIEN LOISEL†

Abstract. The 2-Lagrange multiplier method is a domain decomposition method which can be used to parallelize the solution of linear problems arising from partial differential equations. In order to scale to large numbers of subdomains and processors, domain decomposition methods require a coarse grid correction to transport low frequency information more rapidly between subdomains that are far apart. We introduce two new 2-level methods by adding a coarse grid correction to 2-Lagrange multiplier methods. We prove that if we shrink h (the grid parameter) while maintaining bounded the ratio $\frac{H}{h}$ (where H is the size of the subdomains), the condition number of the method remains bounded. We confirm our analysis with experiments on the HECToR (High-End Computing Terascale Resource) supercomputer. This proves that the new methods scale weakly, opening the door to massively parallel implementations.

Key words. domain decomposition, Schwarz method, partial differential equation, parallel preconditioner, Krylov space, coarse grid, multigrid, 2-Lagrange multiplier, FETI

AMS subject classifications. 35J15, 65N55, 65F08

DOI. 10.1137/140965491

1. Introduction.

The *symmetric* 2*-Lagrange multiplier method* (S2LM) is a linear equation of the form

$$\overbrace{(Q-K)}^{A_{\mathrm{S2LM}}}\lambda = -Qg, \tag{1.1}$$

where Q is symmetric and positive definite, K is an orthogonal projection, g is data, and λ is the unknown. The solution λ can then be used to recover the solution u to a problem $Au = f$ arising from the discretization of an elliptic problem, e.g.,

$$\Delta\tilde{u} = \tilde{f} \text{ in } \Omega \quad \text{and} \quad \tilde{u} = 0 \text{ on } \partial\Omega; \tag{1.2}$$

the domain Ω is an open subset of $\mathbb{R}^2$ or $\mathbb{R}^3$ (see [13] for details).

For solving the problem (1.2), a *domain decomposition* $\Omega = \Omega_1 \cup \cdots \cup \Omega_p \cup \Gamma$ is introduced, where $\Omega_1, \ldots, \Omega_p$ are disjoint (open) *subdomains* and $\Gamma = \Omega \cap \bigcup_{k=1}^{p} \partial\Omega_k$ is the *artificial interface*.

Each subdomain Ω_k gives rise to a "scaled Robin-to-Dirichlet" dense matrix Q_k, and the matrix Q is defined as $Q = \mathrm{blkdiag}(Q_1, \ldots, Q_p)$, corresponding to (1.2). Since Q is block diagonal, calculating the matrix-vector product $Q\lambda$ can be done efficiently in parallel. By comparison, the matrix K is not block diagonal, but instead, K is extremely sparse and hence can be assembled in a parallel sparse matrix format, e.g., in a compressed sparse row (CSR) format, which is the default sparse matrix

*Submitted to the journal's Software and High-Performance Computing section April 16, 2014; accepted for publication (in revised form) February 5, 2015; published electronically April 16, 2015. This work was supported by the Centre for Numerical Algorithms and Intelligent Software (EPSRC EP/G036136/1).
http://www.siam.org/journals/sisc/37-2/96549.html
†Department of Mathematics, Heriot-Watt University, Edinburgh, EH14 4AS, UK (ak411@hw.ac.uk, S.Loisel@hw.ac.uk).

representation for PETSc [1]. Then the computation of the parallel matrix-vector product $K\lambda$ is straightforward.

One of the main goals of domain decomposition is to solve a problem such as (1.2) in parallel. To achieve this objective, one can use a Krylov space solver such as GMRES [17] or MINRES [15] to solve (1.1). It is known [13] that the condition number of (1.1) increases unboundedly when the number of subdomains p increases, which implies poor parallel scaling of the algorithm. In the present paper, we introduce a preconditioner P, the "coarse grid correction," that leads to algorithms with good parallel scaling properties.

This is a major breakthrough for such methods, which have been held back by the lack of a solid theory that enables scaling to large numbers of processors through a coarse grid correction. The only similar work we are aware of is a Fourier analysis [4] of the optimized Schwarz method (OSM) for a model problem on a cylinder where the subdomains are (possibly overlapping) strips. This Fourier analysis is both highly technical and also not applicable to general domains and subdomains and general PDEs. The new theory in the present paper applies to fully general scenarios, including purely algebraic problems as well as problems arising from PDEs on general domains and subdomains. In the present paper, we consider only the nonoverlapping case, while [4] considers both the nonoverlapping and the overlapping cases.

The 2-Lagrange multiplier methods discussed in the present paper were introduced and analyzed in [13], based on a related method introduced in [5]. In the case of two nonoverlapping subdomains, there are two Lagrange multipliers per interface node in Γ, which explains the nomenclature "2-Lagrange multiplier method." The terminology "Lagrange multiplier" is an analogy to the finite element tearing and interconnecting (FETI) methods [6], where the Lagrange multipliers genuinely arise as part of a relaxation of continuity constraints.

The 2-Lagrange multiplier method is known to be closely related to the nonoverlapping OSM [7], which is known to perform well in special cases (e.g., for the spherical Laplacian with two latitudinal subdomains [14]). We also note that cross points (where three or more subdomains meet at a single vertex) have been particularly vexing for the analysis of these methods. Apart from [13] and [8], we are not aware of any analysis of the problem with cross points. We note that the methods introduced in the present paper are able to deal with cross points without any difficulty.

We have two main results. The first is an estimate of the condition number of the S2LM method with its coarse grid preconditioner. This estimate shows that the 2-level S2LM method scales weakly to massively parallel environments. Our second main result provides an estimate that shows that the 2-level nonsymmetric 2-Lagrange multiplier method also scales weakly. In addition to the two main results, we will give strong numerical evidence that the nonsymmetric algorithm works much better in practice.

Our paper is organized as follows. In section 2, we define the basic 2-Lagrange multiplier methods, the coarse grid preconditioner, and the notion of weak scaling. In section 3, we analyze these methods and show that they indeed scale weakly. In section 4, we describe the massively parallel implementation of our methods. In section 5, we provide several numerical experiments that confirm our analysis. We end with some conclusions.

2. 2-Lagrange multiplier methods. We now give a purely algebraic description of the S2LM method.

DEFINITION 2.1 (the matrix A_{S2LM}). *Let $\epsilon > 0$. Assume that the matrix Q is symmetric and positive definite, with the spectrum $\sigma(Q) \subset [\epsilon, 1-\epsilon] \cup \{1\}$. Also assume*

that K is an orthogonal projection. We define A_{S2LM} by (1.1).

Under some conditions, it is known [13] that

$$\frac{1}{\epsilon} = O\left(\sqrt{\frac{H}{h}}\right), \tag{2.1}$$

where H is the Euclidean diameter of a typical subdomain Ω_k, and h is the parameter of the fine grid discretizing (1.2). Therefore, ϵ is a quantity that "scales" in the sense that if we increase the parallelism by shrinking h while keeping H/h bounded, the quantity ϵ remains bounded away from 0.

In the remainder of the present paper, unless noted otherwise, we assume no structure beyond the hypotheses of Definition 2.1. In particular, our analysis applies to systems which arise algebraically and not from the discretization of a boundary value problem.

2.1. Obtaining the S2LM system from (1.2). It is not obvious that the system (1.1) arises naturally from a problem of the form $Au = f$ or (1.2). We now briefly outline a systematic way of obtaining (1.1) from (1.2), and we refer the reader to [13] for details. This "motivational derivation" of (1.1) from (1.2) should be considered the main application of the S2LM scheme.

The main idea of a 2-Lagrange multiplier method is to replace the global problem, (1.2), by the local Robin subproblems,

$$\begin{cases} -\Delta \tilde{u}_k = \tilde{f}_k & \text{in } \Omega_k, \\ \tilde{u}_k = 0 & \text{on } \partial\Omega_k \cap \partial\Omega, \\ (a + D_\nu)\tilde{u}_k = \tilde{\lambda}_k & \text{on } \partial\Omega_k \cap \Gamma, \end{cases} \tag{2.2}$$

where $a > 0$ is the Robin parameter, $k = 0, \ldots, p$, D_ν denotes the directional derivative in the direction of the exterior unit normal ν of $\partial\Omega_k$, and $\tilde{\lambda}_k$ is the Robin flux data imposed on the "artificial interface" $\partial\Omega_k \cap \Gamma$.

By multiplying subproblems (2.2) by a test function $v \in V_k$ and then using Green's formula, we obtain the weak formulation of the local Robin subproblems.

Find $\tilde{u}_k \in V_k$ such that

$$\int_{\Omega_k} \nabla \tilde{u}_k \nabla v dx + a \int_{\partial\Omega_k \cap \Gamma} \tilde{u}_k v dx = \int_{\Omega_k} \tilde{f}_k v dx + \int_{\partial\Omega_k \cap \Gamma} \tilde{\lambda}_k v dx \tag{2.3}$$

holds for all $v \in V_k$, where $V_k = \{v \in H^1(\Omega_k) \mid v = 0 \text{ on } \partial\Omega_k \cap \partial\Omega\}$.

In principle, the term $\int_{\partial\Omega_k \cap \Gamma} \tilde{u}_k v dx$ would give rise to a mass matrix on the artificial interface $\partial\Omega_k \cap \Gamma$. However, it is known (see, e.g., [19, Lemma B.5]) that this mass matrix is spectrally equivalent to the identity matrix by some h factors which have been absorbed into the a parameter.

Hence in matrix form, we obtain a sequence of "Robin problems":

$$\begin{bmatrix} A_{II k} & A_{I\Gamma k} \\ A_{\Gamma I k} & A_{\Gamma\Gamma k} + aI \end{bmatrix} \begin{bmatrix} u_{Ik} \\ u_{\Gamma k} \end{bmatrix} = \begin{bmatrix} f_{Ik} \\ f_{\Gamma k} + \lambda_k \end{bmatrix} \quad \text{for } k = 1, \ldots, p. \tag{2.4}$$

As usual, the indices I and Γ denote the interior and artificial interface nodes, respectively. The vectors $\lambda_1, \ldots, \lambda_k$ are the "Robin data" (one Robin data vector per subdomain). For each subdomain $\Omega_1, \ldots, \Omega_p$ the corresponding restriction matrices $R_1, \ldots, R_p$ are obtained. We have the "partial assembly"

$$A_{Nk} = \begin{bmatrix} A_{II k} & A_{I\Gamma k} \\ A_{\Gamma I k} & A_{\Gamma\Gamma k} \end{bmatrix} \quad \text{and} \quad f_k = \begin{bmatrix} f_{Ik} \\ f_{\Gamma k} \end{bmatrix}, \tag{2.5}$$

leading to the "assembly"

$$A = \sum_k R_k^T A_{Nk} R_k \tag{2.6}$$

(and likewise $f = \sum R_k^T f_k$). The matrix A_{Nk} is obtained by discretizing the bilinear form $\int_{\Omega_k} \nabla u \cdot \nabla v$, and A_{Nk} can be interpreted as the stiffness matrix of a Neumann problem for Ω_k.

We can use Schur complements to eliminate "interior" nodes from (2.4). This leads to the system

$$\overbrace{\begin{bmatrix} S_1 + aI & & \\ & \ddots & \\ & & S_p + aI \end{bmatrix}}^{S+aI} \overbrace{\begin{bmatrix} u_{\Gamma 1} \\ \vdots \\ u_{\Gamma p} \end{bmatrix}}^{u_G} = \overbrace{\begin{bmatrix} g_1 \\ \vdots \\ g_p \end{bmatrix}}^{g} + \overbrace{\begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_p \end{bmatrix}}^{\lambda}, \tag{2.7}$$

where

$$S_k = A_{\Gamma\Gamma k} - A_{\Gamma I k} A_{I I k}^{-1} A_{I \Gamma k} \quad \text{and} \quad g_k = f_{\Gamma k} - A_{\Gamma I k} A_{I I k}^{-1} f_{I k}$$

are the "Dirichlet-to-Neumann maps" and "accumulated right-hand sides."

The vector u_G is the "many-sided" or "multivalued" trace of u—to each interface vertex $\mathbf{x}_i \in \Gamma$, there are several corresponding entries in u_G (one per adjacent subdomain). We used the subscript G by analogy to the notation u_Γ which is normally a single-valued trace. The vector u_G can be used to model a discontinuous function. If all the degrees of freedom of u_G corresponding to $\mathbf{x}_i$ are equal, then we say that u_G is continuous at $\mathbf{x}_i$. We define the square matrix K to be the orthogonal projection whose range is the space of many-sided traces that are continuous on all of Γ; the matrices K and Q are $n \times n$, and the vector u_G is n-dimensional, where n is the number of degrees of freedom along Γ, counting duplicates for each adjacent subdomain. This projection is important because any solution of (1.2) must be continuous across Γ.

The scaled "Robin-to-Dirichlet map" also plays an important role; it is defined by $Q = a(S+aI)^{-1}$; cf. (2.7). Given the matrices Q and K as we have now described, the S2LM system is defined by (1.1), and this system is equivalent to the equation $Au = f$ in the following sense (see [13] for details). If (1.1) is solved, then the solution λ can be used to find $u_1, \ldots, u_p$ using (2.4). These "local solutions" are then guaranteed to meet continuously. By gluing the local solutions together, one obtains u which solves $Au = f$.

An anonymous referee points out that from (1.1), one can derive the equation $K\lambda = au_G$. This is the discrete version of a straightforward continuous phenomenon. If we imagine λ_L and λ_R to be Robin data on the left and right sides of an artificial interface Γ with normal vector ν, then one has formulas such as $\lambda_L = au + D_\nu u$ and $\lambda_R = au + D_{-\nu} u = au - D_\nu u$; note that the direction of the "outward pointing normal" is flipped between the two subdomains. Recalling that K is an averaging operator, one obtains $K\lambda = \frac{1}{2}(\lambda_L + \lambda_R) = au$.

Although this "motivational derivation" of (1.1) from (1.2) is extremely important, it plays no further role in the present paper in the sense that we need only the basic algebraic structure of Definition 2.1; note that Q has the required properties [13].

2.2. Methods that scale weakly. If $1 \notin \sigma(Q)$, one easily checks (see section 3.1) that the spectral condition number $\mathcal{K}(Q-K)$ (the ratio of the singular values) is bounded by $(1-\epsilon)/\epsilon$. On the other hand, if indeed $1 \in \sigma(Q)$, then $\mathcal{K}(Q-K)$ also depends on $\|EK\|$, where E is the orthogonal projection onto the kernel of $I-Q$, I being the identity matrix of appropriate size.

DEFINITION 2.2 (weak scaling). *We say that a method* scales weakly *if the condition number depends only on ϵ and not on the spectral norm $\|EK\|$ (the spectral norm is the largest singular value).*

In domain decomposition, the definition of a method that scales weakly is one where the condition number depends on the ratio H/h, but not on h or H individually. Thus, our definition of weak scaling is justified by the case considered in [13] with ϵ given by (2.1). We now give methods that scale weakly.

DEFINITION 2.3 (2-level 2-Lagrange multiplier (2L2LM) methods). *Let Q, K be as in Definition* 2.1. *Let E be the orthogonal projection onto the kernel of $I-Q$. Assume that $\|EK\| < 1$. We define the* coarse grid preconditioner *as*

$$P = I - EKE, \tag{2.8}$$

leading to the preconditioned matrices

$$A_{\text{2LS2LM}} = P^{-\frac{1}{2}}(Q-K)P^{-\frac{1}{2}} \quad \textit{and} \quad A_{\text{2L2LM}} = P^{-\frac{1}{2}}(I-2K)(Q-K)P^{-\frac{1}{2}}. \tag{2.9}$$

The terminology "2-level" comes from the fact that the action of the preconditioner P^{-1} on a residual can be efficiently computed by projecting onto a coarse grid defined by the subdomains; see (4.5).

Remark 2.4. The matrix P is the "action of $Q-K$ on the range of E." Indeed, if we choose an orthonormal basis such that

$$E = \begin{bmatrix} O & O \\ O & I \end{bmatrix}, \quad Q = \begin{bmatrix} Q_0 & O \\ O & I \end{bmatrix}, \quad \text{and} \quad K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}, \tag{2.10}$$

and where $\sigma(Q_0) \subset (\epsilon, 1-\epsilon)$, then observe that

$$P = \begin{bmatrix} I & O \\ O & I - K_{22} \end{bmatrix} \quad \text{and} \quad Q - K = \begin{bmatrix} Q_0 - K_{11} & -K_{12} \\ -K_{21} & I - K_{22} \end{bmatrix}. \tag{2.11}$$

In other words, the preconditioner P was obtained by "zeroing out" the off-diagonal blocks of $Q-K$ and replacing the top-left block by I.

Remark 2.5. From (2.11) we see that P is symmetric. Furthermore, P is positive definite, provided that $\lambda_{\max}(K_{22}) < 1$. Note that K_{22} is the lower-right block of EKE, and so $\|K_{22}\| \le \|EK\|\|E\| = \|EK\|$. For the elliptic case, $\|EK\| < 1$ is guaranteed (see [13]). Hence the matrix square roots $P^{-1/2}$ are well defined. Instead of computing inverse square roots of P, one would instead implement "left preconditioning":

$$P^{-1}(Q-K) \quad \text{and} \quad P^{-1}(I-2K)(Q-K).$$

These matrices can then be used inside of a suitable implementation of GMRES or similar Krylov space method that uses P as an inner product.

Remark 2.6. The matrix A_{2LS2LM} is symmetric and indefinite, while the matrix A_{2L2LM} is nonsymmetric. We will see in section 3.3 that A_{2LS2LM} and A_{2L2LM} have equivalent condition numbers. Despite this spectral equivalence, we will see in section 4 that GMRES tends to perform better with A_{2L2LM} than on A_{2LS2LM}—this may

be explained by the spectral properties of A_{2L2LM}. Therefore, in practice it may be preferable to use the nonsymmetric matrix A_{2L2LM} instead of the indefinite matrix A_{2LS2LM}.

Remark 2.7. The "coarse space" is the range of E. For the model problem (1.2), the coarse space consists of piecewise constant functions, with one degree of freedom per floating subdomain (subdomains are said to be floating when they do not touch the natural boundary). Since nonzero piecewise constant functions are never continuous, the condition $\|EK\| < 1$ is automatically satisfied (cf. section 2.1). Furthermore, the matrix P can be interpreted as a graph Laplacian for the connectivity graph of the domain decomposition; see [13] for details.

2.3. Connections to the OSM. We now discuss several connections to the OSM. In [13], we discussed the matrix $A_{\mathrm{2LM}} = (M - G)(Q - K)$, which is intimately related to the OSM.

The matrices M and G of [13] correspond roughly to the matrices $I - K$ and K of the present paper. We can make that statement sharper, as follows. After a suitable permutation (see [13, eqs. (2.2), (2.3), (2.6), (2.7)]), the matrices M, G, K have the same block structures. Comparing the blocks of G and K, we find that $\frac{1}{m}G \leq K \leq G$, where m is the maximum degree of the interface vertices and $\cdot \leq \cdot$ denotes the usual symmetric positive definite ordering. Likewise, one finds that the kth block M_k of M has the formula $M_k = \frac{m_k}{m_k - 1}(I - K_k)$, where K_k is the kth block of K and m_k is the degree of the interface vertex corresponding to M_k. As a result, M is spectrally equivalent to $I - K$.

Thus, the nonsymmetric method in [13] is similar to the nonsymmetric method in the present paper in that they are built from "spectrally equivalent components." This analogy is more pronounced in the two-subdomain case where $G = 2K$ and $M = 2(I - K)$ and then $(I - 2K)(Q - K) = \frac{1}{2}A_{\mathrm{2LM}}$. This justifies the definition of A_{2L2LM} as a preconditioning of $A_{\mathrm{2LM}\,*} := (I - 2K)(Q - K)$; cf. (2.9).

When cross points are indeed present, the matrix $2A_{\mathrm{2LM}\,*}$ can be regarded as a generalization of the OSM to the case where the domain decomposition has cross points. The matrix A_{2LM} is also a generalization of the OSM to the situation where there are cross points, but these two generalizations are not precisely the same (except when there are no cross points).

3. Condition number estimates. In the present section, we estimate the condition numbers $\mathcal{K}(A_{\mathrm{2LS2LM}})$ and $\mathcal{K}(A_{\mathrm{2L2LM}})$, and we show that these methods scale weakly.

3.1. The condition number of $A_{\mathrm{2LS2LM}} = P^{-\frac{1}{2}}(Q - K)P^{-\frac{1}{2}}$. Recall the following result.

LEMMA 3.1 (a special case of [11, Corollary 6.3.4]). *Let X and Y be symmetric matrices of the same size. Let $0 < \alpha < \beta < \gamma$ be real numbers. Assume that the spectrum $\sigma(X)$ of X is contained in the interval $[-\alpha, \alpha]$, while $|\sigma(Y)| \subset [\beta, \gamma]$. Then,*

$$|\sigma(X + Y)| \subset [\beta - \alpha, \gamma + \alpha]. \tag{3.1}$$

In order to estimate the condition number of A_{2LS2LM}, first consider the simplest case of $\mathcal{K}(Q - K)$ when $1 \notin \sigma(Q)$ and hence (cf. Definition 2.1) we have the spectral estimate $\sigma(Q) \subset [\epsilon, 1 - \epsilon]$. Since K is an orthogonal projection, we have that $\sigma(K) = \{0, 1\}$. By setting $X := Q - \frac{1}{2}I$ and $Y := -K + \frac{1}{2}I$ and applying (3.1), we find that $|\sigma(Q - K)| \subset [\epsilon, 1 - \epsilon]$. We say that X and Y are obtained by "shifting" Q by $\frac{1}{2}I$.

In the more delicate case where $1 \in \sigma(Q)$, we get that $1/2 \in \sigma(X)$ and $-1/2 \in \sigma(Y)$, which means that there can be cancellation. It is therefore desirable to shift by "more than $\frac{1}{2}I$" in the E component (corresponding to the eigenvalue 1 of Q) in order to obtain $\sigma(X) \subset (-1/2, 1/2)$. In order to estimate the spectrum of Y, one can use the following canonical form for pairs of orthogonal projections.

LEMMA 3.2 (Halmos [9]). *Let E and K be orthogonal projections. There is an orthogonal matrix U which simultaneously block diagonalizes E and K into 1×1 and 2×2 blocks; corresponding blocks of E and K have the same size. If we denote the kth block of the block diagonalized E by E_k, and the kth block of the block diagonalized K by K_k, we further have that*

$$E_k \in \left\{0, 1, \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\right\} \quad \text{and} \quad K_k \in \left\{0, 1, \begin{bmatrix} c_k^2 & c_k s_k \\ c_k s_k & s_k^2 \end{bmatrix}\right\}, \tag{3.2}$$

where $c_k = \cos(t_k) \neq 0$ and $s_k = \sin(t_k) \neq 0$ with real $t_k \in (0, \pi/2)$ for each k.

The ranges of E and K are hyperspaces, and the angles $\{t_k\}$ are the "principal angles" between these two hyperspaces. We are now ready to prove our first main result.

THEOREM 3.3 (condition number and weak scaling of $P^{-\frac{1}{2}}(Q-K)P^{-\frac{1}{2}}$). *Let $0 < \epsilon < \frac{1}{2}$ and assume that Q and K are as in Definition 2.1 and that E and P are as in Definition 2.3. Then, we have the following spectral estimate:*

$$|\sigma(P^{-\frac{1}{2}}(Q-K)P^{-\frac{1}{2}})| \subset \frac{1}{2}\left[\sqrt{4+\epsilon^2} - 2 + \epsilon, \sqrt{4+\epsilon^2} + 2 - \epsilon\right]. \tag{3.3}$$

In particular,

$$\mathcal{K}(P^{-\frac{1}{2}}(Q-K)P^{-\frac{1}{2}}) \leq \frac{\sqrt{4+\epsilon^2}+2-\epsilon}{\sqrt{4+\epsilon^2}-2+\epsilon} \leq \frac{4}{\epsilon}. \tag{3.4}$$

Proof. We compute

$$P^{-\frac{1}{2}}(Q-K)P^{-\frac{1}{2}} = \overbrace{P^{-\frac{1}{2}}(Q-E)P^{-\frac{1}{2}}}^{(Q-E)} + \overbrace{(I-EKE)^{-\frac{1}{2}}(E-K)(I-EKE)^{-\frac{1}{2}}}^{F},$$

where we have used (2.8). According to Lemma 3.2, we have that E, K, F all block diagonalize simultaneously via the orthogonal matrix U. We can further calculate the blocks F_k of F as functions of the blocks E_k and K_k of E and K, and we obtain

$$F_k = \begin{cases} 0 & \text{if } E_k = K_k = 0, \\ -1 & \text{if } E_k = 0 \text{ and } K_k = 1, \\ 1 & \text{if } E_k = 1 \text{ and } K_k = 0, \\ \begin{bmatrix} 1 & -c_k \\ -c_k & -s_k^2 \end{bmatrix} & \text{in the } 2 \times 2 \text{ case,} \end{cases} \tag{3.5}$$

where the case $E_k = K_k = 1$ is excluded by the hypothesis that $\|EK\| < 1$. Let $x > 0$, $y > 0$ be parameters to be selected. We now rewrite $P^{-\frac{1}{2}}(Q-K)P^{-\frac{1}{2}}$ as

$$P^{-\frac{1}{2}}(Q-K)P^{-\frac{1}{2}} = \overbrace{(Q-E) - x(I-E) + yE}^{X} + \overbrace{F + x(I-E) - yE}^{Y}. \tag{3.6}$$

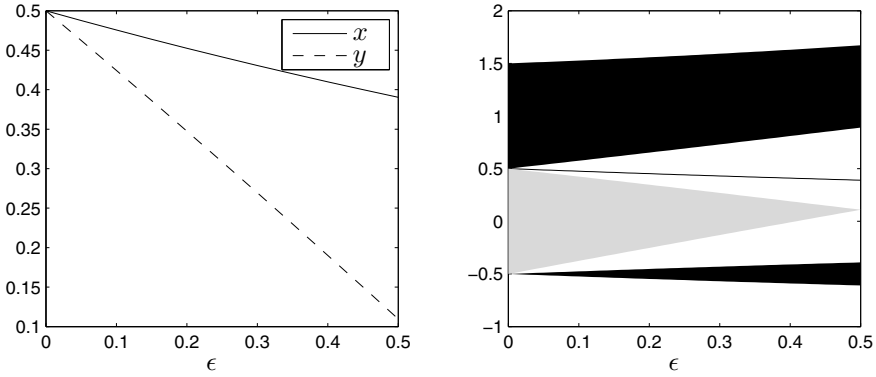

FIG. 1. *Left: The functions $x(\epsilon)$ and $y(\epsilon)$ of* (3.7). *Right: Right-hand sides of* (3.8) *(lightly shaded area) and* (3.12) *(dark areas, including the dark curve $\beta(\epsilon)$).*

We choose

$$x = x(\epsilon) = \frac{\sqrt{4+\epsilon^2}-\epsilon}{4} \quad \text{and} \quad y = y(\epsilon) = \frac{4-3\epsilon-\sqrt{4+\epsilon^2}}{4}; \tag{3.7}$$

cf. Figure 1 (left). We now analyze the spectra of X and Y, using $0 < \epsilon < 1/2$.

We estimate the spectrum of X as

$$\sigma(X) \subset \operatorname{hull}\{\epsilon - x, 1-\epsilon-x, y\} \overset{(3.7)}{=} \left[\frac{5}{4}\epsilon - \frac{1}{4}\sqrt{4+\epsilon^2}, \overbrace{1-\frac{3}{4}\epsilon-\frac{1}{4}\sqrt{4+\epsilon^2}}^{\alpha(\epsilon)}\right], \tag{3.8}$$

where $\operatorname{hull}(Z) := [\inf Z, \sup Z]$ and we have used that the eigenvalues of Q in the "upper-left block" are all in the interval $[\epsilon, 1-\epsilon]$; cf. (2.10).

The eigenvalues of Y can be computed from its block diagonalization, which is computed from (3.5) and (3.6). To the kth block, there corresponds a set $\sigma(Y_k)$ of either one or two eigenvalues, given by

$$\sigma(Y_k) = \begin{cases} \{x\} & \text{if } E_k = K_k = 0, \\ \{x-1\} & \text{if } E_k = 0,\ K_k = 1, \\ \{1-y\} & \text{if } E_k = 1,\ K_k = 0, \\ \{\phi_\pm(c_k^2)\} & \text{in the } 2\times 2 \text{ case}, \end{cases} \tag{3.9}$$

where $\phi_\pm(z) := \frac{1}{2}(z + x - y \pm \sqrt{z^2 + 2(x+y)z + (x+y-2)^2})$ and again the case $E_k = K_k = 1$ is excluded by the hypothesis $\|EK\| < 1$. The function $\phi_+(z)$ is positive (since $y < x$; cf. Figure 1 (left)) and is monotonically increasing since all the coefficients of z are positive. Since $z = c_k^2 = \cos^2 t_k \in [0,1]$, we find that

$$\phi_+(z) \in [\phi_+(0), \phi_+(1)] = \frac{1}{4}\left[3\epsilon + \sqrt{4+\epsilon^2}, \epsilon + 3\sqrt{4+\epsilon^2}\right] \subset (0,\infty); \tag{3.10}$$

cf. (3.7). Now considering $\phi_-(z)$, we find that $\frac{\partial \phi_-}{\partial z} = \frac{1}{2}(1 - (z^2 + 2(x+y)z + (x+y-2)^2)^{-\frac{1}{2}}(z+x+y))$. Using (3.7), we arrive at

$$\frac{\partial \phi_-}{\partial z} = \frac{1}{2}\frac{\overbrace{\sqrt{(1+z-\epsilon)^2+4\epsilon}}^{>1+z-\epsilon>0} - (1+z-\epsilon)}{\sqrt{(1+z-\epsilon)^2+4\epsilon}} > 0,$$

and hence $\phi_-(z)$ is monotonically increasing in z. Therefore,

$$\phi_-(z) \in [\phi_-(0), \phi_-(1)] = \frac{1}{4}\left[-\epsilon - 4 + \sqrt{4+\epsilon^2}, \epsilon - \sqrt{4+\epsilon^2}\right] \subset (-\infty, 0). \tag{3.11}$$

Combining (3.7), (3.9), (3.10), and (3.11), we obtain

$$\begin{aligned}\sigma(Y_k) \subset & \left\{ \overbrace{\frac{1}{4}\sqrt{4+\epsilon^2} - \frac{\epsilon}{4}}^{\beta(\epsilon)} \right\} \cup \left[\frac{3}{4}\epsilon + \frac{1}{4}\sqrt{4+\epsilon^2}, \overbrace{\frac{1}{4}\epsilon + \frac{3}{4}\sqrt{4+\epsilon^2}}^{\gamma(\epsilon)} \right] \\ & \cup \frac{1}{4}\left[-\epsilon - 4 + \sqrt{4+\epsilon^2}, \epsilon - \sqrt{4+\epsilon^2}\right].\end{aligned} \tag{3.12}$$

From (3.8), (3.12), and Figure 1 (right), we find that $\sigma(X) \subset [-\alpha(\epsilon), \alpha(\epsilon)]$ and $|\sigma(Y)| \subset [\beta(\epsilon), \gamma(\epsilon)]$, and (3.3) follows from (3.1). ∎

Remark 3.4. The shifts x, y given by (3.7) were found using the following procedure. According to the discussion at the beginning of the present subsection, it is reasonable to want that $X \approx Q - \frac{1}{2}I$ and $Y \approx -K + \frac{1}{2}I$ when ϵ is small. By inspection of (3.6), we see that this means $x, y \approx \frac{1}{2}$. We hypothesized that good choices of x, y would occur when some of the eigenvalue estimates of X and Y would coincide or equioscillate. We picked a small ϵ and some values of x, y slightly smaller than $\frac{1}{2}$, which seemed to indicate that the eigenvalues $1 - \epsilon - x$ and y of X should coincide, and that the eigenvalues x and $\phi_-(1)$ of Y should cancel: $x + \phi_-(1) = 0$. Solving these two equations for x, y yields (3.7). Having found these values of x, y, we then verified that they indeed produce the estimates (3.3) and (3.4).

Example 3.5. Let $Q(q_1), K(\theta)$, and $P(\theta)$ be defined by

$$Q(q_1) = \begin{bmatrix} q_1 & 0 \\ 0 & 1 \end{bmatrix}, \quad K(\theta) = \begin{bmatrix} c^2 & cs \\ cs & s^2 \end{bmatrix}, \quad P(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & c^2 \end{bmatrix}, \tag{3.13}$$

where $c = \cos\theta$ and $s = \sin\theta$ for some real parameter θ. Note that $\|EK\| = |s|$. Setting $z := c^2$, we find that $\sigma(P^{-\frac{1}{2}}(\theta)(Q(q_1) - K(\theta))P^{-\frac{1}{2}}(\theta)) = \{\psi_\pm(q_1, z)\}$, where

$$\psi_\pm(q_1, z) = \frac{1}{2}\left(1 + q_1 - z \pm \sqrt{z^2 - 2(1+q_1)z + q_1^2 - 2q_1 + 5}\right). \tag{3.14}$$

For $q_1 = 1 - \epsilon \in (0, 1)$, letting $z \to 0$ shows that (3.3) is sharp.

The condition number estimate (3.4) depends only on ϵ, and hence, according to our definition, the 2-level S2LM method is scalable.

3.2. The condition number of $A_{\text{2L2LM}} = P^{-\frac{1}{2}}(I - 2K)(Q - K)P^{-\frac{1}{2}}$. Note that

$$A_{\text{2L2LM}} = \overbrace{P^{-\frac{1}{2}}(I - 2K)P^{\frac{1}{2}}}^{Z} A_{\text{2LS2LM}}. \tag{3.15}$$

LEMMA 3.6. *Assume that* $\|EK\| < 1$ *and let* $Z = P^{-\frac{1}{2}}(I - 2K)P^{\frac{1}{2}}$. *Then,*

$$\mathcal{K}(Z) \le \frac{\sqrt{2}+1}{\sqrt{2}-1} < 5.83. \tag{3.16}$$

Proof. We block diagonalize Z using Lemma 3.2 and (2.8). The blocks Z_k of Z are as follows:

$$Z_k = \begin{cases} 1 & \text{if } E_k = K_k = 0, \\ -1 & \text{if } E_k = 0 \text{ and } K_k = 1, \\ 1 & \text{if } E_k = 1 \text{ and } K_k = 0, \\ \begin{bmatrix} 1 - 2c_k^2 & -2c_k \\ -2s_k^2 c_k & 1 - 2s_k^2 \end{bmatrix} & \text{in the } 2 \times 2 \text{ case,} \end{cases} \tag{3.17}$$

where the case $E_k = K_k = 1$ is excluded by the hypothesis that $\|EK\| < 1$. Replacing $s_k^2 = 1 - c_k^2$, we compute the singular values $\Sigma(Z_k)$ of Z_k to obtain

$$\Sigma(Z_k) \subset \left\{ 1, \sqrt{1 + 2c_k^6 \pm 2c_k^3 \sqrt{1 + c_k^6}} \right\}. \tag{3.18}$$

Optimizing $c_k \in [0, 1]$ in (3.18) for the largest possible condition number (which occurs at $c_k = 1$) gives (3.16). ∎

We now prove our second main result.

THEOREM 3.7 (condition number of $A_{2\text{L}2\text{LM}} = P^{-\frac{1}{2}}(I - 2K)(Q - K)P^{-\frac{1}{2}}$). *Let $0 < \epsilon < \frac{1}{2}$ and assume that Q and K are as in Definition* 2.1 *and that E and P are as in Definition* 2.3. *Then, we have the following condition number estimate:*

$$\mathcal{K}(A_{2\text{L}2\text{LM}}) < \frac{23.32}{\epsilon}. \tag{3.19}$$

Proof. We use the submultiplicativity of condition numbers on (3.15) combined with the estimates (3.4) and (3.16). ∎

According to our definition, $A_{2\text{L}2\text{LM}}$ scales weakly.

3.3. Motivation for the nonsymmetric system. In section 2.3, we discussed the relationship between $A_{2\text{L}2\text{LM}}$ and the OSM, which is one motivation for the study of $A_{2\text{L}2\text{LM}}$. In the present section, we discuss another reason to prefer $A_{2\text{L}2\text{LM}}$ over $A_{2\text{LS}2\text{LM}}$ related to the spectral properties of these matrices.

The matrix $A_{2\text{LS}2\text{LM}}$ is symmetric but typically indefinite (when K is a nontrivial projection) despite the fact that the initial problem (1.2) was symmetric and positive definite. Because $A_{2\text{LS}2\text{LM}}$ is indefinite, it cannot be used with the conjugate gradient method (CG) [10], but a method such as GMRES or MINRES (which also has a two-term recurrence) can be used. The performance of CG depends on the square root of the condition number, whereas the performance of MINRES depends on the condition number (without a square root). For instance, in the elliptic case, the matrix $A_{2\text{LS}2\text{LM}}$ has the condition number $O(\sqrt{H/h})$ (using (3.4), provided that (2.1) holds); thus using GMRES or MINRES on $A_{2\text{LS}2\text{LM}}$ may have a performance comparable to using CG on, e.g., additive Schwarz with minimal overlap, which has a condition number of $O(H/h)$ [19, Theorem 3.13].

Note that, in exact arithmetic, GMRES and MINRES applied to a symmetric indefinite matrix such as $A_{2\text{LS}2\text{LM}}$ produce the same iterates. In machine arithmetic, MINRES suffers from a "loss of orthogonality" [16, p. 195], which means that its performance is usually worse than that of GMRES. We have briefly discussed MINRES because it is the go-to solver for symmetric indefinite systems, but in order to avoid these numerical complications, we focus on GMRES for the remainder of the present paper.

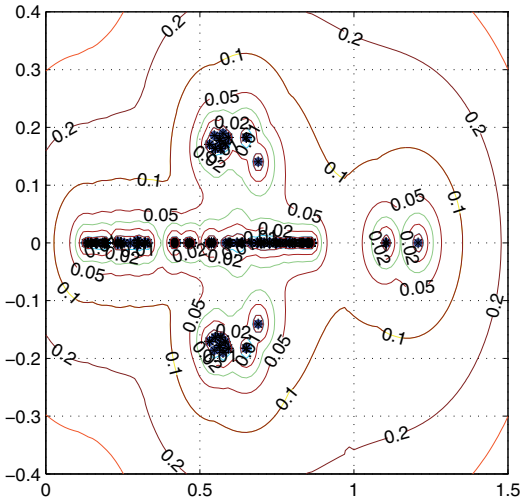

FIG. 2. *Spectrum and pseudospectrum of* A_{2L2LM}.

The matrix A_{2L2LM} is nonsymmetric.[1] We have no analysis linking the condition number $\mathcal{K}(A_{\text{2L2LM}})$ to the performance of GMRES, but our numerical experiments suggest that GMRES applied to A_{2L2LM} is much more efficient than GMRES applied to A_{2LS2LM}. Note that the transformation from A_{2LS2LM} to A_{2L2LM} by left-multiplying by the reflection matrix $I - 2K$ is very similar to the positive definite reformulation of saddle point problems, which is experimentally known to be better for iterative solvers, even though there is no analysis [2]. Our experiments suggest the performance of GMRES may depend on $\sqrt{\mathcal{K}(A_{\text{2L2LM}})} = O((H/h)^{1/4})$, a significant improvement over $O(\sqrt{H/h})$. In Figure 2, we have plotted the spectrum and pseudospectrum of A_{2L2LM}, computed numerically from an example for the problem (1.2) on the unit square with $h = 1/32$ and $H = 1/4$. In principle, the performance of GMRES can be analyzed by bounding the set of eigenvalues and using potential theory; see [3]. In this context, our eigenvalue set consists of complex eigenvalues of mild moduli (these will not significantly impact convergence), and real or nearly real eigenvalues that approach zero. Hence, the convergence behavior suggested by Figure 2 is expected to be comparable to the case where the eigenvalues are in some interval $[\delta, 1.5]$, plus a few iterations to take care of the mild complex eigenvalues. We confirm this good behavior with numerical experiments in section 5.3.

4. Massively parallel implementation. We present the parallel implementation of the 2LS2LM and 2L2LM methods with cross points and with the coarse grid preconditioner (2.8) described by systems

$$P^{-1}(Q - K)\lambda = -P^{-1}Qg, \tag{4.1}$$

$$P^{-1}(I - 2K)(Q - K)\lambda = -P^{-1}(I - 2K)Qg, \tag{4.2}$$

respectively. We implemented these methods in C using the PETSc [1] library. Our code works for general domains Ω and subdomains Ω_i of arbitrary shapes.

[1] This is true unless Q and K commute. This is unlikely to happen for a "random" matrix Q but can exceptionally happen, e.g., if there are two subdomains and $Q_1 = Q_2$.

Our objective was to create a 2-Lagrange multiplier "black-box solver" that takes as an input a parallel distributed matrix which holds the "splitting" (2.6). Then the solver functions algebraically on the given input information in order to solve numerically problem (1.2), using either the 2LS2LM or the 2L2LM methods. The PETSc type `MATIS` is used as an input for our solver since it can efficiently encode and store (2.6).

Moreover, we have implemented our own parallel mesh generation and partitioning algorithm, which has been used along with Triangle, a two-dimensional mesh generator and Delaunay triangulator [18], and gives as an output (2.6).

4.1. Mesh generation and assembly of local Neumann problems. We start from a seed mesh $\mathcal{T}_0$ that describes the general geometry of our problem. Then by further refinement of $\mathcal{T}_0$ a new coarse mesh $\mathcal{T}_H$ is created. Each triangle of $\mathcal{T}_H$ becomes a subdomain and is assigned to a unique processor. The user provides the number m of vertices to be generated on each edge of the coarse mesh $\mathcal{T}_H$, and the mesh is refined accordingly in order to create the desired refined mesh $\mathcal{T}_h$. The fine mesh can in general be very large, so it is created on a per subdomain basis.

Since $\mathcal{T}_h$ is not globally assembled, only the local numbering of the nodes is known to each processor. In order for each processor to acquire the global numbering of its nodes, without any communication cost, we designed the following algorithm.

For each subdomain Ω_i of the fine mesh we can compute the number of vertices that belong to it. Each vertex of the fine mesh ν_i is labeled with an integer $i = 1, \ldots, n$. Each processor has a corresponding subdomain Ω_i with neighbors Ω_j. For the subdomain Ω_i and its neighbors Ω_j such that $j < i$ the fine mesh is created; cf. Figure 3. The information in Figure 3 is sufficient to compute the global labels ℓ_i of the vertices $\nu_i \in \Omega_i$, without assembling the global fine mesh and without any message passing interface (MPI) communication.

Next, we assign each of the fine vertices to a single owner subdomain in the following way. Vertices that lie in the interior of a subdomain Ω_i are assigned to subdomain Ω_i. Vertices along an edge $\partial\Omega_i \cap \partial\Omega_j$ but not at a cross point are assigned to the subdomain $\Omega_{\min(i,j)}$. Vertices at a cross point $v^* \in \partial\Omega_i \cap \partial\Omega_j \cap \cdots \cap \partial\Omega_k$ are assigned to the subdomain $\Omega_{\min(i,j,\ldots,k)}$.

Each subdomain Ω_i consists of vertices $v_i \in \overline{\Omega_i}$, some of which have been assigned to subdomain Ω_j and others to the neighboring subdomains with smaller numbering. This way, the precise ownership of each $v_i \in \overline{\Omega_i}$ can be computed locally without the need to assemble the global fine mesh.

Once the global labels of the vertices of the fine mesh are computed, a "local to globlal mapping" which specifies the binary restriction matrices R_i is defined. Because R_i restricts to Ω_i, only the labels of the vertices in $\overline{\Omega_i}$ are required, and hence no communication is needed in order to assemble R_i.

Likewise, the Neumann matrices A_{Ni} are computed and assembled as `seqaij` matrices without communication. The resulting objects $\{R_i, A_{Ni}\}$ form the PETSc distributed matrix of type `MATIS`.

4.2. The 2LS2LM and 2L2LM black-box solver. The solver takes as an input a matrix of type `MATIS` that contains the information about the local Neumann problems A_{Ni} and the restriction matrices R_i. The matrices K, Q, and P in (4.1) and (4.2) are implemented. Matrix K is assembled as a parallel `mpiaij` matrix, in a compressed row storage matrix format. Since P, Q are dense matrices, they are not assembled explicitly but instead are implemented as PETSc "matrix-free" matrices by defining the matrix vector products $P\lambda$ and $Q\lambda$, respectively.

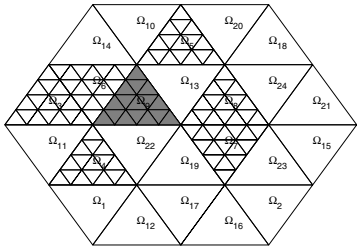

FIG. 3. *The processor that has been assigned the "gray" subdomain Ω_9 refines all the neighboring subdomains Ω_j with $j < 9$.*

The matrix K is assembled as a parallel sparse matrix and is defined from the following product of matrices:

$$K = W R_\Gamma R_\Gamma^T, \quad \text{where } R_\Gamma = \begin{bmatrix} R_{\Gamma_1} \\ \vdots \\ R_{\Gamma_p} \end{bmatrix}, \tag{4.3}$$

$W = (\mathrm{diag}(R_\Gamma R_\Gamma^T \mathbf{1}))^{-1}$, and $\mathbf{1}$ corresponds to a vector of ones.

The matrix Q is implemented in a matrix-free form. Since Q is a block diagonal matrix of submatrices Q_k, it is implemented as the matrix-vector product $\lambda_k \mapsto Q_k \lambda_k$. This product can be computed by solving the local sparse Robin problem,

$$\begin{bmatrix} A_{Ilk} & A_{I\Gamma k} \\ A_{\Gamma Ik} & A_{\Gamma\Gamma k} + aI \end{bmatrix} \begin{bmatrix} u_{Ik} \\ u_{\Gamma k} \end{bmatrix} = \begin{bmatrix} 0 \\ \lambda_k \end{bmatrix}, \tag{4.4}$$

for each λ_k of the multivalued trace vector λ.

Remark 4.1. The solution λ of either (4.1) or (4.2) is a many-sided trace with one function value per artificial interface point per subdomain. The vector λ is a PETSc parallel vector object, the rows of which are distributed in such a way that the indices of the same domain are assigned to a single processor.

The coarse grid preconditioner P defined in (2.8) can be assembled in a black-box manner. Let $\mathbf{1}_{n_k}$ denote the n_k-dimensional column vector of ones, where n_k is the size of A_{Nk}; see (2.5). Since $\mathbf{1}_{n_k}$ spans $\ker A_{Nk}$, we can detect the floating subdomains by checking whether the product $A_{Nk}\mathbf{1}_{n_k} = 0$, with some tolerance.

Then we are able to produce the basis of the coarse space as

$$J := \mathrm{blkdiag}\left(\frac{1}{\sqrt{n_{\Gamma 1}}} \mathbf{1}_{n_{\Gamma 1}}, \ldots, \frac{1}{\sqrt{n_{\Gamma p}}} \mathbf{1}_{n_{\Gamma p}} \right),$$

where $n_{\Gamma k}$ is the number of vertices on the artificial interface $\partial\Omega_k \cap \Gamma$. This orthonormal basis for the range of E gives the formula $E = JJ^T$, allowing us to implement P^{-1} in a matrix-free way. We now use the block notation (2.11). Note that in this basis, $J = \left[\begin{smallmatrix} O \\ I \end{smallmatrix}\right]$. We find that

$$P^{-1} = \begin{bmatrix} I & O \\ O & (I - K_{22})^{-1} \end{bmatrix} = \begin{bmatrix} I & O \\ O & (I - J^T K J)^{-1} \end{bmatrix} = \overbrace{\begin{bmatrix} I & O \\ O & O \end{bmatrix}}^{I-E} + \overbrace{\begin{bmatrix} O & O \\ O & (I - J^T K J)^{-1} \end{bmatrix}}^{J(I-J^T K J)^{-1} J^T}.$$

Thus,

$$P^{-1} = I - JJ^T + J(I - J^T K J)^{-1} J^T. \tag{4.5}$$

The matrices J and J^T and the $p \times p$ coarse problem $L = I - J^T K J$ are assembled explicitly. Given λ, we compute the p-dimensional vector $\lambda_c = J^T \lambda$, and we gather its entries on a single processor as a sequential vector. Then, we solve the coarse problem $Lu_c = \lambda$ using an LU decomposition. The sequential vector u_c is scattered to all processors, and finally we get the desired output from

$$P^{-1}\lambda = \lambda - J\lambda_c + Ju_c.$$

We also define the matrices for $(Q - K)$ and $(I - 2K)(Q - K)$ in a matrix-free form with the corresponding matrix-vector product operations, $\lambda \mapsto (Q - K)\lambda$ and $\lambda \mapsto (I - 2K)(Q - K)\lambda$.

In order to solve the systems (4.1), (4.2) we use the parallel generalized minimal residual Krylov subspace method `KSPGMRES` provided by the PETSc library with the preconditioner P given by (4.5).

Finally, once the solution λ of either (4.1) or (4.2) is obtained, the solution of the global problem $Au = f$ is recovered locally by solving (2.4). This is due to the fact that the final step of solving (2.4) requires only the local part λ_k of the parallel vector λ.

5. Numerical experiments. We have four sets of numerical experiments. In the first set (the "algebraic case"), we generate random matrices Q and K to validate (3.4). In the second set of experiments (the "elliptic case"), we use the model problem (1.2) with various values of h and H to validate the estimates (2.1), (3.4), (3.19). In the third set of experiments, we measure the performance of GMRES and GMRES(10) on A_{2LS2LM} and A_{2L2LM}. Finally, in the fourth set, in order to check the scalability of our massively parallel implementation described in section 4, we present some large-scale experiments performed on the HECToR (High-End Computing Terascale Resource) supercomputer.

5.1. Algebraic case. Our first series of numerical experiments (Figure 4) consists of generating random matrices Q and K and verifying the estimate (3.4). For each plot, we generate 1000 random matrices Q and K. For each matrix Q, we fix the ambient dimension n, the parameter $\epsilon > 0$, and the number k of eigenvalues of Q that are less than 1. We then set the smallest eigenvalue to ϵ and the largest eigenvalue smaller than 1 to $1 - \epsilon$, and the remaining $k - 2$ eigenvalues are picked randomly and uniformly in the interval $[\epsilon, 1 - \epsilon]$. The matrix Q is then taken to be the corresponding diagonal matrix.

We also generate K randomly as follows. First, we fix the dimension m of the range of K. Then we generate a matrix V of dimension $n \times m$ whose columns are orthonormal, and we set $K = VV^T$. The matrix V is generated randomly with the MATLAB command `orth(rand(n,m)-0.5)`.

Each such experiment produces a condition number for $P^{-\frac{1}{2}} A_{\text{S2LM}} P^{-\frac{1}{2}}$, which is plotted as a dot against the value of ϵ in Figure 4. Although the resulting probability distribution of points does depend on the parameters (k, m, n), we find that the estimate (3.4) (plotted as a solid line in Figure 4) holds and seems to be sharp.

5.2. Elliptic case. Our second set of experiments is on the model problem (1.2). We discretized the unit square with a regular grid of parameter h. We partitioned this

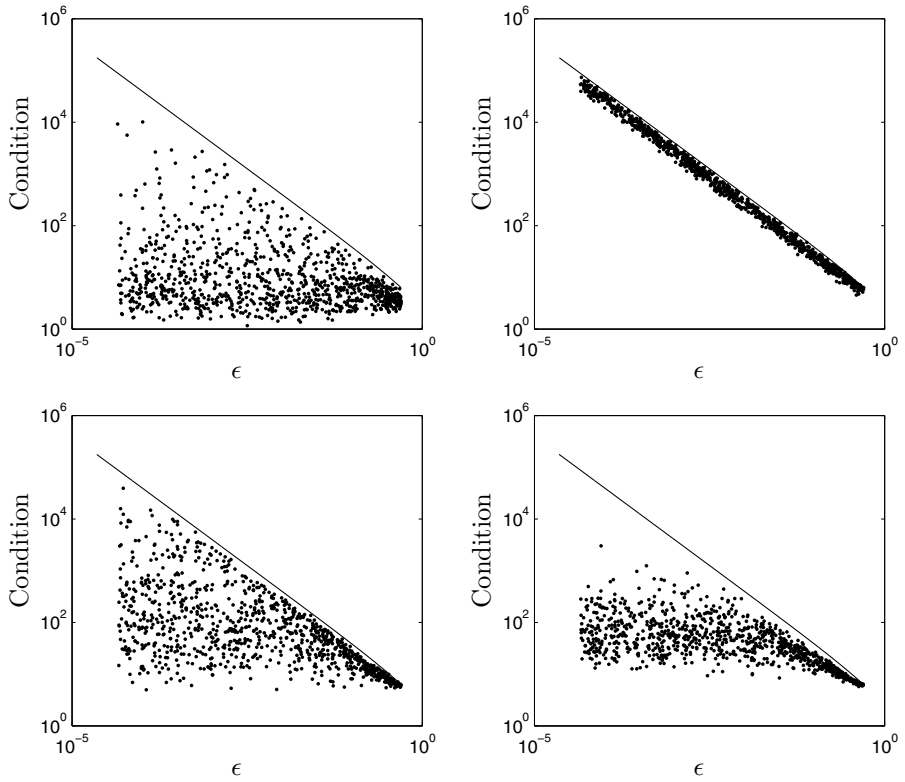

FIG. 4. *Condition numbers of A_{2LS2LM} for random choices of Q and K for various values (m, n, k) (dots) compared to* (3.4) *(solid curve). Top left: $n = 4, k = 3, m = 2$. Top right: $n = 8, k = 4, m = 4$. Bottom left: $n = 15, k = 8, m = 7$. Bottom right: $n = 30, k = 18, m = 15$.*

square into square subdomains of side H, with up to 256 subdomains. We then assembled the matrices A_{2LS2LM} and A_{2L2LM} and computed the condition numbers using the MATLAB command `cond`.[2] The results are summarized in Tables 1 (symmetric case) and 2 (nonsymmetric case).

The estimate (2.1) implies that the condition numbers along the diagonals of Tables 1 and 2 should be bounded, which appears to be the case (this is "weak scaling"). Furthermore, the estimate (2.1) implies that moving one column to the right ought to increase the condition number by a factor of $\sqrt{2} \approx 1.4$, which is also approximately verified. Indeed, the relative increases for the last column of Table 1 compared to the penultimate column are $1.38, 1.36, 1.34$. The corresponding ratios in Table 2 are $1.41, 1.40, 1.39$.

We highlight the fact that this set of numerical experiments includes many cross points and a large number of floating subdomains. Our new 2-level method is able to deal with these challenging situations without difficulty and with good scaling properties.

We also note that our estimate (3.19) of the condition number of A_{2L2LM} is

[2]We found that `cond(X)` gives much less accurate results when `X` is a sparse matrix. This is because `cond` then uses the approximate condition number estimate `condest(X)`. In order to obtain more accurate results, we stored A_{2LS2LM} and A_{2L2LM} as dense matrices.

TABLE 1
Condition numbers for the 2*-level* symmetric 2*-Lagrange multiplier matrix* A_{2LS2LM} *for the model problem* (1.2).

	h				
	0.1250	0.0625	0.0313	0.0156	0.0078
$H = 0.2500$	7.1927	9.6909	13.2186	18.1121	25.0399
$H = 0.1250$		9.3567	12.2031	16.4196	22.3863
$H = 0.0625$			9.8847	12.8234	17.2216

TABLE 2
Condition numbers for the 2*-level* nonsymmetric 2*-Lagrange multiplier matrix* A_{2L2LM} *for the model problem* (1.2).

	h				
	0.1250	0.0625	0.0313	0.0156	0.0078
$H = 0.2500$	6.1508	8.4162	11.6954	16.3073	22.9120
$H = 0.1250$		7.3623	9.9804	13.8537	19.3739
$H = 0.0625$			7.6480	10.3564	14.3824

5.83 times worse than the estimate (3.4), but this is not borne out in our numerical experiments. Indeed, the matrix A_{2L2LM} appears to be better conditioned than the matrix A_{2LS2LM}. Our estimate of the condition number of A_{2L2LM} was obtained using the "rough" idea of the submultiplicativity of condition numbers, which is apparently very conservative in the present situation.

5.3. Performance with GMRES and GMRES(10). Our third set of experiments (cf. Figure 5) consists of using the GMRES and restarted GMRES(10) iterations on the matrices of section 5.2, where the initial residual is a column vector of ones. We now briefly discuss these results, starting with GMRES. Since A_{2LS2LM} is symmetric, we can use standard theory to estimate the convergence of GMRES (which in this case is equivalent to MINRES). A worst-case bound is [12, p. 291]

$$\frac{\|r_k\|_2}{\|r_0\|_2} = O\left(\left(\frac{\mathcal{K}-1}{\mathcal{K}+1}\right)^{k/2}\right). \tag{5.1}$$

This is quite a slow convergence, and this estimate is known not to be sharp when the spectrum exhibits some asymmetry about the origin. Indeed, we see that, when h is large, GMRES on A_{2LS2LM} performs even better than

$$\frac{\|r_k\|_2}{\|r_0\|_2} = O\left(\left(\frac{\mathcal{K}-1}{\mathcal{K}+1}\right)^{k}\right), \tag{5.2}$$

where $k = 0, 1, \ldots$ is the iteration count, r_k is the corresponding residual, and the condition number $\mathcal{K} = 12.8$ from Table 1 was used. As can be observed in Figure 5, the linear estimate (5.2) is very pessimistic when h is large. As a result, the scalability of the algorithm is only apparent when h is very small.

As mentioned in section 3.3, the matrix A_{2L2LM} has much better spectral properties. The present experiments suggest that the correct linear estimate for the con-

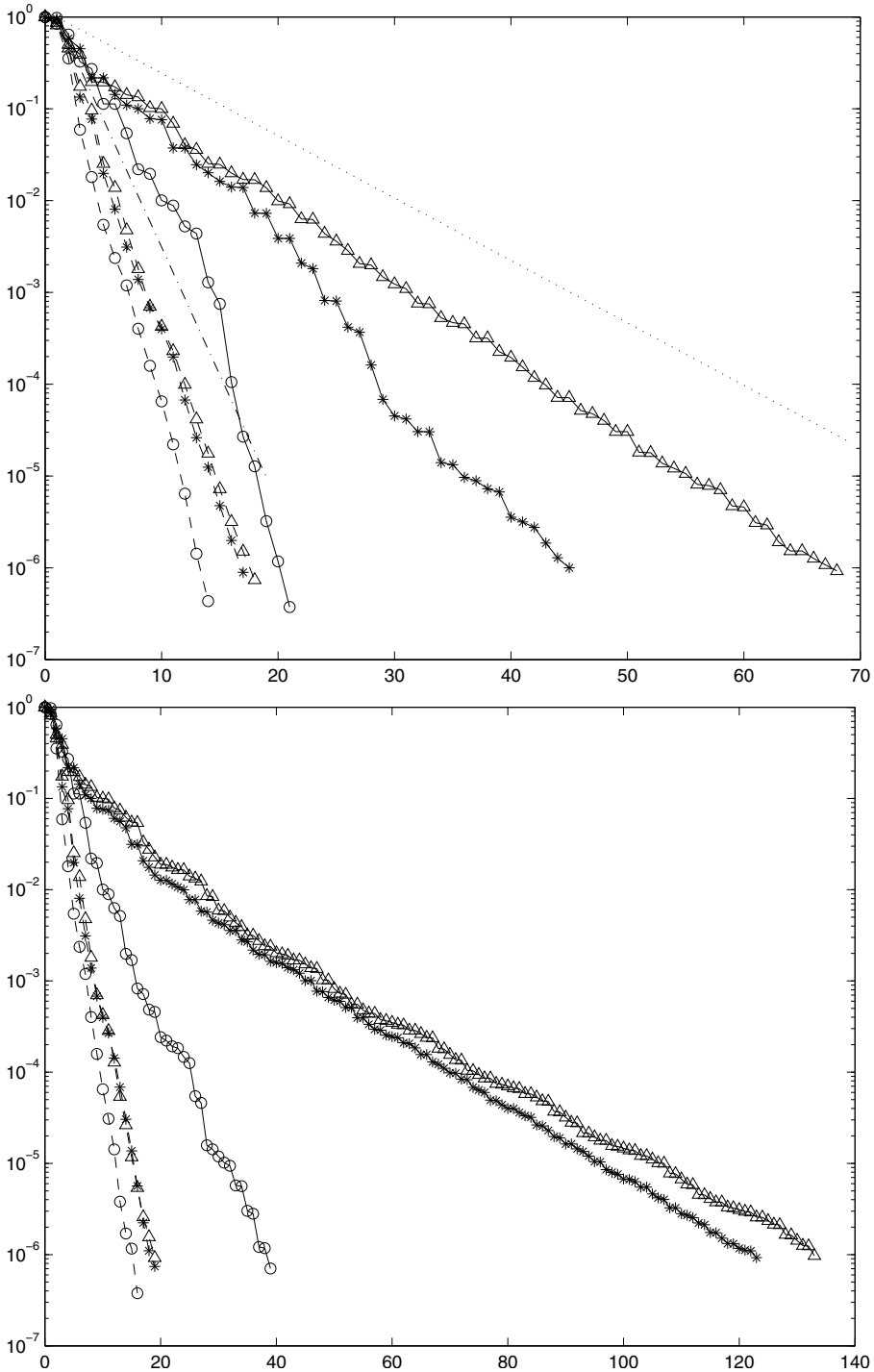

FIG. 5. *Convergence of the relative residual norm in the GMRES (top) and GMRES*(10) *(bottom) iterations (to a relative tolerance of* 10^{-6}*) with grid parameters* $h = \frac{1}{4}, H = \frac{1}{16}$ *(circles),* $h = \frac{1}{4}, H = \frac{1}{32}$ *(stars), and* $h = \frac{1}{4}, H = \frac{1}{64}$ *(triangles). The solid lines correspond to* A_{2LS2LM}*, while the dashed lines correspond to* A_{2L2LM}*. In the top figure, the dotted line is* (5.2)*, and the dot-dashed line is* (5.3).

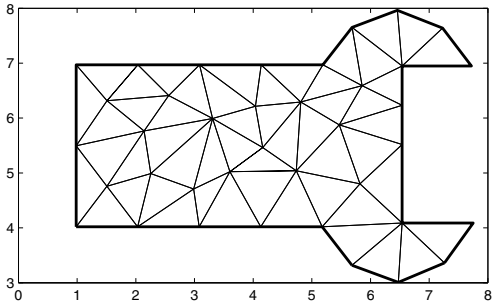

FIG. 6. *Wrench-shaped domain* Ω.

vergence of GMRES applied to A_{2L2LM} is

$$O\left(\left(\frac{\sqrt{\mathcal{K}}-1}{\sqrt{\mathcal{K}}+1}\right)^{k}\right). \tag{5.3}$$

(The value $\mathcal{K} = 10.4$ from Table 2 was used.) This may be related to the fact that the spectrum of A_{2L2LM} is essentially a positive interval plus some complex eigenvalues of mild moduli; cf. Figure 2.

We now turn to the GMRES(10) experiments (Figure 5, bottom). The restarted GMRES algorithm can be used when the storage requirements of the full GMRES algorithm are too high. We have used the GMRES(10) algorithm, which restarts every tenth iteration. Although the performance of A_{2LS2LM} appears scalable, the iteration counts are now much higher. By contrast, the matrix A_{2L2LM} is scalable in all cases, and the iteration counts are nearly the same as in the full GMRES algorithm (less than 20 in all cases).

5.4. Large scale experiments on HECToR supercomputer. In this subsection we present some results, in terms of iteration counts and the wall clock time, that correspond to the massively parallel implementation of A_{2L2LM} and A_{2LS2LM} described in section 4. We solve the problem (1.2) with the constant function $f = 1$ as a right-hand side, where Ω is the wrench-shaped domain in Figure 6. In this set of experiments in order to solve systems involving A_{2L2LM} or A_{2LS2LM} with the coarse grid preconditioner (2.8), we have used the generalized minimal residual method `KSPGMRES`, with 1e-7 relative tolerance and 1e-6 absolute tolerance, respectively.

The number of grid points in the domains varies from 10^5 to 10^8. Moreover, the domains are partitioned into 51–3264 subdomains. The experiments were performed on the HECToR supercomputer, where one subdomain is assigned to each processor. The results, in terms of iteration counts and the wall clock time, are presented in Tables 3–6. The full code for our A_{2L2LM} or A_{2LS2LM} solver implementation can be found online at https://bitbucket.org/modios/matis_2lm.

We see that the nonsymmetric method 2L2LM produces very moderate iteration counts (103 iterations in the very worst case), while the symmetric method 2LS2LM produces many more iterations. In principle this suggests that one should use the nonsymmetric method to obtain better performance. However, the higher number of iterations is not always reflected in the wall clock time. This is partially because the HECToR supercomputer requires a significant amount of time to distribute our

TABLE 3
Iteration counts for 2LS2LM.

# Procs.	Number of grid points ≈ 10^5	$4 \cdot 10^5$	10^6	$7 \cdot 10^6$	$2 \cdot 10^7$
51	489	859	1369	1438	–
204	445	597	888	1363	–
816	316	510	616	947	1898
3264	–	–	–	617	1108

TABLE 4
Wall clock time (seconds) for 2LS2LM.

# Procs.	Number of grid points ≈ 10^5	$4 \cdot 10^5$	10^6	$7 \cdot 10^6$	$2 \cdot 10^7$
51	27s	30s	54s	167s	–
204	27s	28s	35s	62s	–
816	43s	53s	61s	83s	172s
3264	–	–	–	610s	1055s

TABLE 5
Iteration counts for 2L2LM.

# Procs.	Number of grid points ≈ 10^5	$4 \cdot 10^5$	10^6	$7 \cdot 10^6$	$2 \cdot 10^7$	10^8
51	43	50	58	72	–	–
204	44	51	64	77	92	–
816	41	48	55	67	81	103
3264	–	–	–	48	67	83

TABLE 6
Wall clock time (seconds) for 2L2LM.

# Procs.	Number of grid points ≈ 10^5	$4 \cdot 10^5$	10^6	$7 \cdot 10^6$	$2 \cdot 10^7$	10^8
51	25s	26s	30s	48s	–	–
204	26s	27s	27s	29s	51s	–
816	28s	28s	29s	31s	36s	68s
3264	–	–	–	90s	118s	135s

tasks to all the nodes in the cluster (for smaller problems, this is essentially all of our running time). However, for the largest problems, we gain one order of magnitude in the wall clock time, simply by using the nonsymmetric method.

The scaling properties are also better in the nonsymmetric method. For the symmetric method, going from $7 \cdot 10^6$ to $2 \cdot 10^7$ grid points increases the iteration counts by factors of $1898/947 \approx 2.004$ and $1108/617 \approx 1.796$. By comparison, the nonsymmetric method with the same number of processors increases the iteration counts only by factors of 1.209 and 1.396, respectively, so we have much better scaling properties from the 2L2LM method than the 2LS2LM method.

The communication overheads for PETSc on HECToR were significant, and we can see in some cases that problems of a certain size require a longer wall clock time when processors are added. Although we made some effort to optimize this, we

concluded that significant engineering efforts would be required to extract the most performance from this hardware.

5.5. Discussion. In view of the numerical experiments of section 5.3, we recommend the use of $A_{2\text{L}2\text{LM}}$ over $A_{2\text{LS}2\text{LM}}$, in combination with GMRES or restarted GMRES. The matrix $A_{2\text{L}2\text{LM}}$ requires many fewer GMRES iterations than the matrix $A_{2\text{LS}2\text{LM}}$ to converge, possibly because of the spectral properties of $A_{2\text{L}2\text{LM}}$ and $A_{2\text{LS}2\text{LM}}$. Furthermore, the matrix $A_{2\text{L}2\text{LM}}$ is more robustly scalable. A restarted version of GMRES also works well with $A_{2\text{L}2\text{LM}}$, and less well with $A_{2\text{LS}2\text{LM}}$.

6. Conclusions. We have introduced new 2-level 2-Lagrange multiplier methods featuring a coarse grid correction. We have estimated the condition numbers of our new methods and shown that our new methods scale weakly, which is an important factor when choosing a domain decomposition method for parallel computing. We have shown that our condition number estimates are sharp. Our algebraic estimates apply to the elliptic case for general domains and subdomains with general elliptic PDEs, as described in section 2.1; see also [13] and references therein. Cross points pose no special difficulty. The theory has been confirmed by numerical experiments. In combination with GMRES (with or without restart), the nonsymmetric method is superior. Numerical experiments suggest that the performance of the nonsymmetric algorithm combined with GMRES depends on $(H/h)^{1/4}$.

REFERENCES

[1] S. Balay, J. Brown, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, *PETSc: Portable, Extensible Toolkit for Scientific Computation*, http://www.mcs.anl.gov/petsc (2013).

[2] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.

[3] T. A. Driscoll, K.-C. Toh, and L. N. Trefethen, *From potential theory to matrix iterations in six steps*, SIAM Rev., 40 (1998), pp. 547–578.

[4] O. Dubois, M. J. Gander, S. Loisel, A. St-Cyr, and D. B. Szyld, *The optimized Schwarz method with a coarse grid correction*, SIAM J. Sci. Comput., 34 (2012), pp. A421–A458.

[5] C. Farhat, A. Macedo, M. Lesoinne, F.-X. Roux, F. Magoulès, and A. de La Bourdonnaie, *Two-level domain decomposition methods with Lagrange multipliers for the fast iterative solution of acoustic scattering problems*, Comput. Methods Appl. Mech. Engrg., 184 (2000), pp. 213–239.

[6] C. Farhat and F.-X. Roux, *A method of finite element tearing and interconnecting and its parallel solution algorithm*, Internat. J. Numer. Methods Engrg., 32 (1991), pp. 1205–1227.

[7] M. J. Gander, *Schwarz methods in the course of time*, Electron. Trans. Numer. Anal., 31 (2008), pp. 228–255.

[8] M. J. Gander and F. Kwok, *Best Robin parameters for optimized Schwarz methods at cross points*, SIAM J. Sci. Comput., 34 (2012), pp. A1849–A1879.

[9] P. R. Halmos, *Two subspaces*, Trans. Amer. Math. Soc., 144 (1969), pp. 381–389.

[10] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436.

[11] R. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[12] J. Liesen and Z. Strakoš, *Krylov Subspace Methods: Principles and Analysis*, Oxford University Press, Oxford, UK, 2013.

[13] S. Loisel, *Condition number estimates for the nonoverlapping optimized Schwarz method and the 2-Lagrange multiplier method for general domains and cross points*, SIAM J. Numer. Anal., 51 (2013), pp. 3062–3083.

[14] S. Loisel, J. Côté, M. J. Gander, L. Laayouni, and A. Qaddouri, *Optimized domain decomposition methods for the spherical Laplacian*, SIAM J. Numer. Anal., 48 (2010), pp. 524–551.

[15] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
[16] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
[17] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[18] J. R. Shewchuk, *Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator*, in Applied Computational Geometry Towards Geometric Engineering, Springer, Berlin, Heidelberg, 1996, pp. 203–222.
[19] A. Toselli and O. B. Widlund, *Domain Decomposition Methods—Algorithms and Theory*, Springer Ser. Comput. Math. 34, Springer, Berlin, Heidelberg, 2005.