SIAM J. Numer. Anal.
Vol. 34, No. 4, pp. 1616–1639, August 1997

015

THE SPECTRAL PROJECTION DECOMPOSITION METHOD FOR ELLIPTIC EQUATIONS IN TWO DIMENSIONS[*]

P. GERVASIO[†], E. OVTCHINNIKOV[‡], AND A. QUARTERONI[§]

Abstract. The projection decomposition method (PDM) is invoked to extend the application area of the spectral collocation method to elliptic problems in domains compounded of rectangles. Theoretical and numerical results are presented demonstrating the high accuracy of the resulting method as well as its computational efficiency.

Key words. domain decomposition, spectral methods, elliptic problems

AMS subject classifications. 65N55, 65N35

PII. S0036142994265334

Introduction. Spectral methods represent a relatively new approach to the numerical solution of partial differential equations which appeared more recently than the widely used finite difference and finite element methods (FEMs). The first attempt of a systematic analysis was undertaken by D. Gottlieb and S. Orszag in their celebrated monograph [16], while a more comprehensive study appeared only 10 years later in [7], where spectral methods were shown to be a viable alternative to other numerical methods for a broad variety of mathematical equations, and particularly those modeling fluid dynamical processes. As a matter of fact, spectral methods feature the property of being accurate up to an arbitrary order, provided the solution to be approximated is infinitely smooth and the computational domain is a Cartesian one. Moreover, if the first condition is not fulfilled, the method automatically adjusts to provide an order of accuracy that coincides with the smoothness degree of the solution (measured in the scale of Sobolev spaces). This is undoubtedly a potential superiority enjoyed by these methods over traditional finite differences and the h-version of the FEMs. On the other hand, when the computational domain $\Omega \subset \mathbb{R}^d$ has a complex shape, finite elements have the advantage of being more flexible in fitting the domain boundaries by arbitrarily complex grids. However, spectral methods can still be applied provided that the boundary $\partial\Omega$ is regular enough to allow the domain to be partitioned into subregions that can be easily mapped into a reference d-dimensional cube. This approach has led to the development of the *spectral domain decomposition* methods based on the Galerkin method (the so-called “spectral element method”; see [22], [18]) or the collocation method using Gauss–Lobatto points on each subdomain (see [20], [7, Chapter 13] and [25]). Decompositions by overlapping subdomains constructed by enlarging the original nonoverlapping ones have also been considered (e.g., [30], [6], [23]). However, here the situation is not as simple as

[*] Received by the editors March 25, 1994; accepted for publication (in revised form) December 9, 1995. This research has been carried out with the support of the Sardinian Regional Authorities while the first and second authors were in residence at the Centre for Advanced Studies Research and Development in Sardinia -CRS4-. The first and third authors also acknowledge the support of Ministero Università Ricerca Scientifica Tecnologica (M.U.R.S.T.) through “Fondi 40%.”

http://www.siam.org/journals/sinum/34-4/26533.html

[†] Department of Electronics, University of Brescia, via Branze 38, 25123 Brescia, Italy (gerva@pitagora.bs.unicatt.it).

[‡] Centre for Techno-Mathematics and Scientific Computing, University of Westminster, 115 New Cavendish Street, London, W1M 8JS, England (ovtchie@westminster.ac.uk).

[§] Departimento di Matematica, Politecnico, di Milano, via Bonardi, 9, 20133 Milano, Italy (aq@mate.polimi.it).

for low-order, local methods (finite differences or finite elements) since the Gaussian grids on the enlarged subdomains do not include as a subset those on the original subdomains, and therefore one has to apply interpolation procedures which can be computationally very expensive. To avoid the interpolation, the enlarged subdomains can be taken as unions of several original subdomains, as proposed by Pavarino in [23]. The convergence behavior of such a method is quite satisfactory, but the extra computational work is still very large due to the large amount of overlap.

In this paper we consider an extension of the spectral collocation method to elliptic problems in polyrectangular domains (i.e., domains compounded of rectangles) based on the novel nonoverlapping domain decomposition algorithm—the so-called *projection decomposition method* (PDM) [3, 21]. In this method the equation expressing the continuity of conormal derivative across the interface between the subdomains, the so-called *interface Steklov–Poincaré equation*, is solved by the Galerkin projection method. The use of a piecewise-polynomial *well-conditioned basis* leads to a well-conditioned linear system for the coordinates of the Galerkin approximation in this basis which can be solved effectively by the conjugate gradient (CG) method with the convergence rate independent of the system dimension. This enables us to achieve a desired accuracy of the approximate solution by increasing the degree N of polynomials used for approximation in the subdomains and on the interface without slowing down the convergence rate of the CG iterations. Besides, the combination of the piecewise-polynomial Galerkin approximation on the interface and the spectral collocation method in the subdomains results in a high accuracy of the numerical solutions.

We stress that in the method proposed here a prescribed accuracy of approximation is achieved *by increasing the degree of polynomials used while keeping the number of subdomains fixed* (cf. p-version of FEM): the use of preconditioning and advanced mapping techniques, e.g., transfinite blending, allows effective exploitation of such a method in a wide class of geometries.

Thus, *one of the main objectives of this paper is to coordinate the PDM with a well-conditioned piecewise-polynomial basis used for reducing the problem in a polyrectangular domain to a sequence of subproblems in rectangles and the spectral collocation method used for solving those subproblems.* We show that the resulting algorithm is very effective and provides a solution that approximates the exact one with the spectral accuracy. We show also that the convergence rate of the subdomain iterations is independent of N and depends mildly on the number of subdomains even though no coarse grid solver is called into play.

An outline of the paper is as follows. Sections 1 and 2 introduce the statement of the problem and its spectral collocation approximation in the case of a rectangular domain. In section 3 we briefly describe the PDM with a well-conditioned polynomial basis: this description summarizes that from [3, 21] and is included here for the reader's convenience. Then in section 4 we show how this method can be coupled with the spectral collocation method inside each subdomain, yielding a projection decomposition–spectral collocation (PD–SC) method. Finally, in section 5 we demonstrate the numerical performance of our PD–SC method in several test cases. An extensive comparison is carried out with two other domain decomposition methods, notably

(i) the Dirichlet–Neumann method with dynamical choice of the optimal acceleration parameter; and

(ii) the overlapping Schwarz method in its additive form: the CG iterations on the spectral element matrix are preconditioned using the additive Schwarz method with coarse grid solver induced by piecewise bilinear finite elements on the Gauss–

Lobatto grid; this approach proposed in [15] is similar to the one in [24] and it makes use of an overlapping region of variable thickness.

The superior efficiency of our method over both alternatives is clearly demonstrated, leading us to the conclusion that this method represents a very efficient tool for solving elliptic boundary value problems in complex two-dimensional (2D) domains with the spectral accuracy.

1. The problem.

Consider the following boundary value problem:

$$\left\{ \begin{array}{rl} -\Delta u + \kappa u = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{array} \right. \tag{1}$$

where Δ is the Laplace operator, Ω is a bounded two-dimensional domain with piecewise-smooth Lipschitz boundary $\partial\Omega$, κ is a nonnegative bounded function, and f is a function in $L^2(\Omega)$.

We denote by

(i) $L^2(\Omega)$ the space of measurable functions $u : \Omega \to \mathbb{R}$ such that $\int_\Omega |u(x,y)|^2 \mathrm{d}\Omega < +\infty$ endowed with the norm

$$||u||_{L^2(\Omega)} = \left(\int_\Omega |u(x,y)|^2 \mathrm{d}\Omega \right)^{1/2} ; \tag{2}$$

(ii) $L^\infty(\Omega)$ the space of measurable functions $u : \Omega \to \mathbb{R}$ such that $|u(x,y)|$ is bounded outside a set of measure zero; the norm in $L^\infty(\Omega)$ is

$$||u||_{L^\infty(\Omega)} = \operatorname*{ess\,sup}_{(x,y)\in\Omega} |u(x,y)|; \tag{3}$$

(iii) $H^1(\Omega)$ the space of functions $v \in L^2(\Omega)$ whose first-order distributional derivatives belong to $L^2(\Omega)$, endowed with the norm

$$||v||_{H^1(\Omega)} = \{||v||^2_{L^2(\Omega)} + ||\nabla v||^2_{L^2(\Omega)}\}^{1/2}; \text{ and} \tag{4}$$

(iv) $H^1_0(\Omega)$ the subspace of $H^1(\Omega)$ of the functions whose trace at the boundary is zero, i.e.,

$$H^1_0(\Omega) = \{v \in H^1(\Omega) \ : \ v|_{\partial\Omega} = 0\} \tag{5}$$

endowed with the norm

$$||v||_1 = ||\nabla v||_{L^2(\Omega)}. \tag{6}$$

The variational formulation of (1) reads

$$\begin{array}{c} \text{find } \ u \in H^1_0(\Omega) \text{ such that} \\ a(u,v) = b(v) \qquad \forall v \in H^1_0(\Omega) \end{array} \tag{7}$$

with

$$a : H^1(\Omega) \times H^1(\Omega) \to \mathbb{R}, \quad b : H^1(\Omega) \to \mathbb{R}, \tag{8}$$

$$a(u,v) = \int_\Omega (\nabla u \cdot \nabla v + \kappa uv)\, d\Omega \ , \quad b(v) = \int_\Omega fv\, d\Omega. \tag{9}$$

It is easy to see that there exist positive constants $\hat{a}_0, \check{a}_0$, and b_0 such that

$$|a(u,v)| \le \hat{a}_0 \parallel u \parallel_1 \parallel v \parallel_1, \quad a(u,u) \ge \check{a}_0 \parallel u \parallel_1^2, \quad |b(v)| \le b_0 \parallel v \parallel_1; \tag{10}$$

therefore problem (7) has a unique solution owing to the Riesz theorem.

2. The spectral collocation method. Here we briefly describe the spectral collocation method (SCM) [7, 4, 28] for the problem (7) with inhomogeneous boundary data on a rectangle Ω^*:

$$\left\{ \begin{array}{ll} Lu \equiv -\Delta u + \kappa u = f & \text{in } \Omega^*, \\ u = g & \text{on } \partial\Omega^*, \end{array} \right. \tag{11}$$

where $f \in C^0(\Omega^*)$ and $g \in C^0(\partial\Omega^*) \cap H^{1/2}(\partial\Omega^*)$. The bilinear form associated to the operator L is the same as in (9). The weak formulation of the inhomogeneous problem (11) reads

$$\text{find } u \in V_g \ : \ a(u,v) = \int_{\Omega^*} fv \ d\Omega^* \quad \forall v \in V, \tag{12}$$

where $V_g = \{v \in H^1(\Omega^*) \ : \ (v-g) \in H_0^1(\Omega^*)\}$ and $V = H_0^1(\Omega^*)$. To solve problem (12) we apply the SCM at Legendre–Lobatto nodes described below.

We denote by $\xi_k^{N^*}$ and $\gamma_k^{N^*}$ ($k = 0, \ldots, N^*$) the nodes and the weights of the Gauss–Lobatto quadrature formula of order $(2N^* - 1)$ on the interval $(-1, 1)$ with respect to the weight function $\omega(x) \equiv 1$. We remind the reader that (e.g., [9])

$$\xi_0^{N^*} = -1, \ \xi_{N^*}^{N^*} = 1, \ \text{while } \xi_k^{N^*} \ \ (k = 1, \ldots, N^* - 1) \text{ are the zeros of } L'_{N^*}. \tag{13}$$

Here L_{N^*} is the Legendre polynomial in $(-1, 1)$ of degree N^* and the weights $\gamma_k^{N^*}$ have the following expression:

$$\gamma_k^{N^*} = \frac{2}{N^*(N^*+1)} \frac{1}{[L_{N^*}(\xi_k)]^2}, \quad k = 0, \ldots, N^*. \tag{14}$$

Given two natural numbers N_x and N_y we denote by $\mathbb{P}_N(\Omega^*)$ the space of algebraic polynomials on Ω^* of degree N_x and N_y with respect to the variables x and y, respectively. We denote by (x_i, y_j) (for $i = 0, \ldots, N_x$ and $j = 0, \ldots, N_y$) the nodes on Ω^* obtained by the Cartesian product of $\xi_i^{N_x}$ and $\xi_j^{N_y}$, while $\omega_{ij} = \gamma_i^{N_x} \gamma_j^{N_y} \frac{\text{meas}(\Omega^*)}{4}$.

We will make use of the following notations:

$(u,v)_{N,\Omega^*} = \sum_{i,j=1}^{N_x,N_y} u(x_i,y_j)v(x_i,y_j)\omega_{ij}$ is the discrete inner product in $L^2(\Omega^*)$ on the Gauss–Lobatto–Legendre (GLL) nodes $(x_i, y_j) \in \Omega^*$;

$(u,v)_{N,\partial\Omega^*}$ is the GLL quadrature formula for the integration of uv on each side of $\partial\Omega^*$;

$V_N = \{p \in \mathbb{P}_N(\Omega^*) : p = 0 \text{ on } \partial\Omega^*\}$.

The discrete counterpart of the bilinear form (9) is

$$a_N(u,v) = (\nabla u, \nabla v)_{N,\Omega^*} + (\kappa u, v)_{N,\Omega^*} \tag{15}$$

and it can be shown that $\forall u_N, v_N \in \mathbb{P}_N(\Omega^*)$

$$a_N(u_N, v_N) = (-\Delta u_N + \kappa u_N, v_N)_{N,\Omega^*} + \Big(\frac{\partial u_N}{\partial n}, v_N\Big)_{N,\partial\Omega^*} \tag{16}$$

(e.g., [28, Chapter 6]).

The approximation of (12) by the SCM reads

$$\text{find } u_N \in \mathbb{P}_N(\Omega^*) : a_N(u_N, v_N) = (f, v_N)_N \quad \forall \ v_N \in V_N \tag{17}$$

and $u_{N} = g$ at all collocation nodes lying on $\partial\Omega^*$. Let $\{\chi^j_{N^*}(x)\}_{j=0,N^*}$ be the set of the Lagrange polynomials defined on the interval (a,b) such that $\chi^j_{N^*} \in \mathbb{P}_{N^*}(a,b)$, $\chi^j_{N^*}(x_k) = \delta_{kj}$, and let $\{\chi^{ij}_{N}(x,y)\}_{\substack{i=0,N_x\\ j=0,N_y}}$ be the set of the Lagrange polynomials defined on Ω^* such that $\chi^{ij}_N \in \mathbb{P}_N(\Omega^*)$, $\chi^{ij}_N(x,y) = \chi^i_{N_x}(x)\chi^j_{N_y}(y)$. It can be derived from (16) and (17), by setting $v_N = \chi^{ij}_N$, that the spectral collocation approximation to the solution u of the problem (12) is the algebraic polynomial $u_N \in \mathbb{P}_N(\Omega^*)$ such that

$$\left\{\begin{array}{rl} Lu_N = f & \text{at any } (x_i,y_j) \in \Omega^*, \\ u_N = g & \text{at any } (x_i,y_j) \in \partial\Omega^*. \end{array}\right. \tag{18}$$

The spectral collocation method is known to be very accurate provided that the solution of the problem (11) is smooth enough (see, e.g., [4], [7], [28], and the references therein). However, in the form presented above it is restricted to problems in a rectangle. In order to generalize this method without loss of accuracy to more general geometries, we apply the domain decomposition technique presented in the subsequent section.

3. The PDM. Suppose that the domain Ω is partitioned into nonoverlapping subdomains $\Omega_i,\ i = 1,\dots,m$:

$$\overline{\Omega} = \bigcup_{i=1}^{m} \overline{\Omega}_i, \quad \Omega_i \cap \Omega_j = \emptyset \text{ for } \ i \neq j. \tag{19}$$

All subdomains are assumed to be rectangles, each one sharing either a whole side or a vertex with each neighbor. We denote by $\Omega_0 = \cup_{k=1}^m \Omega_k$ the union of the subdomains and by $\Gamma = \Omega \setminus \Omega_0$ the union of subdomain interfaces, briefly referred to below as the interface. Following the standard domain decomposition technique based on the concept of the Poincaré–Steklov operator (see, e.g., [2, 27]; for the alternative approach based on the *subspace decomposition technique*, see, e.g., [12] and the references therein) we decouple problem (7) into m *local* boundary value problems, one in each subdomain, and a problem on the interface between the subdomains as follows (see also [5], where a similar technique was used). We represent the solution u of problem (7) as the sum

$$u = u_0 + v, \tag{20}$$

where $u_0 \in H^1_0(\Omega_0)$ is the solution of the problem

$$a(u_0,w) = b(w) \qquad \forall w \in H^1_0(\Omega_0), \tag{21}$$

while $v \in H^1_0(\Omega)$ is the solution of the problem

$$a(v,w) = b(w) - a(u_0,w) \equiv g(w) \qquad \forall w \in H^1_0(\Omega). \tag{22}$$

Problem (21) yields upon each subdomain Ω_k a problem like (7) (i.e., an elliptic problem with homogeneous Dirichlet condition). The problem (22) in its turn can be reduced to a problem for the trace $\varphi = u_{|\Gamma}$ of the solution u on Γ through the following well-known derivation used in the theory of the Poincaré–Steklov operators [2, 27].

Let us introduce the following subspace of $H^1_0(\Omega)$:

$$H_{harm} = \{w \in H^1_0(\Omega) : a(w,z) = 0 \qquad \forall z \in H^1_0(\Omega_0)\}. \tag{23}$$

From (21) and (22) it follows that $a(v,w) = 0 \ \forall w \in H_0^1(\Omega_0)$, i.e., $v \in H_{harm}$, and therefore problem (22) can be reformulated as

$$\begin{aligned} &\text{find } \ v \in H_{harm} \text{ such that} \\ &a(v,w) = b(w) \qquad \forall w \in H_{harm}. \end{aligned} \tag{24}$$

We note that when $\kappa = 0$ in (9), functions from H_{harm} are harmonic in each subdomain. Below we call these functions "harmonic" even when κ is different from zero.

From the well-known results of the theory of traces (see, e.g., [14, 17, 29]) it follows that H_{harm} is isomorphic to the space

$$H_0^{1/2}(\Gamma) = \{\varphi = w_{|\Gamma} : w \in H_0^1(\Omega)\} \tag{25}$$

provided with the norm

$$\| \varphi \|_{H_0^{1/2}(\Gamma)} = \inf_{w_{|\Gamma}=\varphi} \| w \|_{H_0^1(\Omega)} . \tag{26}$$

Introducing the operator $E_\Gamma : H_0^{1/2}(\Gamma) \rightarrow H_{harm}$, the inverse of the trace operator $Tu = u_{|\Gamma}$ restricted to H_{harm}, we eventually rewrite problem (24) as the following variational problem in $H_0^{1/2}(\Gamma)$:

$$\begin{aligned} &\text{find } \ \varphi \in H_0^{1/2}(\Gamma) \text{ such that} \\ &a(E_\Gamma\varphi, E_\Gamma\psi) = b(E_\Gamma\psi) \qquad \forall \psi \in H_0^{1/2}(\Gamma). \end{aligned} \tag{27}$$

Below this problem is referred to as the Poincaré–Steklov problem.

A specific feature of PDM is the use of the Galerkin method for the approximation of the Poincaré–Steklov problem. A system $\Phi^n = \{\varphi_i^n\}_{i=1,\nu_n}$ of linearly independent functions $\varphi_i^n : \Gamma \rightarrow \mathbb{R}$ is introduced (cf. [1]) and the problem (24) is approximated by the finite dimensional one:

$$\begin{aligned} &\text{find } \ v^n \in H_n = \text{span } \{E_\Gamma\Phi^n\} \text{ such that} \\ &a(v^n, w) = b(w) \qquad \forall w \in H_n. \end{aligned} \tag{28}$$

Here the index n refers to the number of functions corresponding to each segment $\Gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$ of Γ, while ν_n denotes the global number of functions on Γ. Functions φ_i^n below are called *interface basis functions* and their "harmonic" extensions $w_i^n = E_\Gamma\varphi_i^n$ are called *"harmonic" basis functions.*

Representing v^n as a linear combination of the "harmonic" basis functions

$$v^n = \sum_{j=1}^{\nu_n} a_j w_j^n, \tag{29}$$

we obtain from (28) the following linear system:

$$\hat{S}\hat{a} = \hat{b}, \tag{30}$$

where

$$\hat{S} = [a(w_j^n, w_i^n)]_{i=1,\nu_n}^{j=1,\nu_n}, \quad \hat{a} = [a_j]_{j=1,\nu_n}, \quad \hat{b} = [b(w_i^n)]_{i=1,\nu_n}. \tag{31}$$

From (10) it follows that $\hat{S}$ is symmetric and positive definite; therefore we can solve the system (31) using the CG method.

We note that from the Green formula related to the bilinear form a we have the following expressions for the terms $b(w_i^n)$ and $a(v^n, w_i^n)$ appearing in the calculation of the residual for (30):

$$b(w_i^n) = -\sum_{k=1}^{m} \int_{\partial\Omega_k \cap \Gamma} \varphi_i^n \frac{\partial u_0}{\partial n_k}\, d(\partial\Omega_k), \tag{32}$$

$$a(v^n, w_i^n) = \sum_{k=1}^{m} \int_{\partial\Omega_k \cap \Gamma} \varphi_i^n \frac{\partial v^n}{\partial n_k}\, d(\partial\Omega_k), \tag{33}$$

where $\frac{\partial}{\partial n_k}$ is the normal derivative on $\partial\Omega_k$ and the integrals are interpreted as duality pairing between $H_0^{1/2}(\Gamma)$ and its dual. Therefore for the implementation of the PDM we need not calculate the "harmonic" basis functions (see also section 3.1 for a comment on this issue).

3.1. Summary of the algorithm.

The algorithm induced by the PDM can be summarized as follows (a more detailed description will be provided in section 4.3).

Step1 Solve the Dirichlet problem (21) on each subdomain in order to obtain $u_0 \in H_0^1(\Omega_0)$. Then compute the right-hand side $\hat{b} = [b(w_i^n)]_{i=1,\nu_n}$ for the system (30) as indicated in (32).

Step2 Solve (30) using a suitable iterative method (e.g., the CG method). At each iteration l of the CG method we have to apply the matrix $\hat{S}$ to a vector which represents the descent direction of the CG (see section 4.3). Precisely, given a vector $\hat{p}^l = [p_i^l]_{i=1}^{\nu_n}$ we need to compute the vector $\hat{q}^l = \hat{S}\hat{p}^l = [q_i^l]_{i=1}^{\nu_n}$, where, according to (33),

$$q_i^l = \sum_{k=1}^{m} \int_{\partial\Omega_k \cap \Gamma} \varphi_i^n \frac{\partial p^l}{\partial n_k} d(\partial\Omega_k), \qquad i = 1, \dots, \nu_n, \tag{34}$$

and the function $p^l \in H_n$ is defined as $p^l = \sum_{i=1}^{\nu_n} p_i^l w_i^n$. We point out that the computation of p^l does not require the knowledge of the $\{w_i^n\}$, as this function can be obtained by solving the following m independent Dirichlet problems: for $k = 1, \dots, m$

$$\begin{cases} -\Delta p^l + \kappa p^l = 0 & \text{in } \Omega_k, \\ p^l = 0 & \text{on } \partial\Omega_k \backslash \Gamma, \\ p^l = p_\Gamma^l & \text{on } \partial\Omega_k \cap \Gamma, \end{cases} \tag{35}$$

where $p_\Gamma^l = \sum_{i=1}^{\nu_n} p_i^l \varphi_i^n$.

Since each subdomain Ω_k is a rectangle, we can use the SCM for solving the subdomain problems (21), (35).

3.2. Convergence analysis of PDM.

Three main sources of the error of the above PDM–SCM algorithm are (1) the SCM error (i.e., the error resulting from the spectral approximation in the subdomains); (2) the CG error (i.e., the error of the approximate solution of (30) with the CG algorithm: in exact arithmetic this error vanishes after no more than ν_n steps; however, in practical calculations one usually stops iterations earlier); and (3) the PDM error (i.e., the approximation error of the

Galerkin method (28)). Since the first two sources are well studied, here we focus our analysis on the third one.

THEOREM 1. *Let the system $\Phi^n = \{\varphi_i^n\}_{i=1,\nu_n}$ consist of functions that are algebraic polynomials of degree not greater than n on each segment $\Gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$ of the interface Γ. Then the PDM approximation $\{u^n = u_0 + v^n,\ n \geq 1\}$, where u_0 is the solution of* (21) *and v^n is that of* (28), *converges to the exact solution u of the problem* (7) *and the following convergence estimate holds:*

$$\| u - u^n \|_{H^1(\Omega)} \leq \frac{C}{n^{s-1}} \max_{k=1,\dots,m} \| u \|_{H^s(\Omega_k)} \tag{36}$$

provided that $u_{|\Omega_k} \in H^s(\Omega_k)$ for some $s \geq 2$. Furthermore, if the sequence Φ^n is well conditioned in $H_0^{1/2}(\Gamma)$ (in the sense of Mikhlin [19]*; see also* [3]*), i.e., there exist two constants $\rho_0,\ \rho^0$ independent of n such that*

$$\rho_0 \sum_{i=1}^{\nu_n} b_i^2 \leq \left\| \sum_{i=1}^{\nu_n} b_i \varphi_i^n \right\|^2_{H_0^{1/2}(\Gamma)} \leq \rho^0 \sum_{i=1}^{\nu_n} b_i^2 \tag{37}$$

for all $[b_i]_{i=1,\nu_n} \in \mathbb{R}^{\nu_n}$, then the condition number of the system (30) *is bounded by the constant $\rho^0 \hat{a}_0 (\rho_0 \check{a}_0)^{-1}$.*

Proof. From (24) and (28) we have

$$a(v - v^n, h) = 0 \quad \forall h \in H_n;$$

therefore

$$a(v - v^n, v - v^n) = \min_{h \in H_n} a(v - h, v - h). \tag{38}$$

Introducing the notation

$$\varepsilon_n = \min_{h \in H_n} \| v - h \|_1 \tag{39}$$

we obtain from (10) and (38)

$$\varepsilon_n \leq \| v - v^n \|_1 \leq \varepsilon_n \left(\frac{\hat{a}_0}{\check{a}_0} \right)^{1/2}, \tag{40}$$

and therefore PDM approximations v^n converge to the solution v of problem (24) if and only if $\varepsilon_n \to 0$.

From the definition of H_{harm} we have immediately that

$$\forall h \in H_{harm} \quad a(h,h) \leq a(z,z) \quad \forall z \in H_0^1(\Omega) : z_{|\Gamma} = h_{|\Gamma}.$$

Using this property together with (10), we have for any $h_n \in H_n$

$$\| v - h_n \|_1^2 \leq \check{a}_0^{-1} a(v - h_n, v - h_n) \leq \check{a}_0^{-1} a(z,z) \leq \check{a}_0^{-1} \hat{a}_0 \| z \|_1^2$$
$$\forall z \in H_0^1(\Omega) : z_{|\Gamma} = (v - h_n)_{|\Gamma}$$

since $v - h_n \in H_{harm}$. We recall that v is such that $v_{|\Gamma} = u_{|\Gamma}$. Let us take $z = u - I_n u,\ h_n \in H_n : h_{n|\Gamma} = (I_n u)_{|\Gamma}$, where in each Ω_k $I_n u$ is an algebraic polynomial of degree less than or equal to n with respect to each coordinate x and y that interpolates

u at the $(n+1)^2$ GLL nodes (we recall that we consider a conforming rectangular decomposition). Using well-known estimates for polynomial interpolation in Sobolev spaces (see, e.g., [7, Chapter 9] and [4]) and the best approximation property (38) of the Galerkin method, we obtain the estimate (36).

Finally, the estimate for the condition number of system (30) follows from (10) and (37). $\square$

3.3. The construction of a well-conditioned basis. Here we briefly describe the algorithm for constructing a well-conditioned piecewise-polynomial basis for the PDM; a detailed description and analysis of the construction outlined below can be found in [3, 21].

We start with the domain $\Omega = (-1,1)\times(-1,1)$ decomposed into four square subdomains.

Consider two sequences of functions $\{\psi_k^0(x)\}$ and $\{\varphi_k^0(x)\}$:

$$\psi_k^0(x) = x(x-\tfrac{1}{k})(x-\tfrac{2}{k})\cdots(x-1),$$

$$\varphi_{2k-1}^0(x) = \begin{cases} \psi_k^0(x), & x>0,\\ -\psi_k^0(-x), & x<0,\end{cases} \tag{41}$$

$$\varphi_{2k}^0(x) = \begin{cases} \frac{\psi_k^0(x)}{x}, & x>0,\\ \varphi_{2k}^0(-x), & x<0,\end{cases}$$

with which we construct the sequences $\{u_k^0\}$ and $\{v_k^0\}$:

$$\begin{array}{lll} & u_k^0, v_k^0 \in H_{harm}(D),\ D=(0,1)\times(0,1): & \\ u_k^0(x,0) = \psi_k^0(x), & u_k^0(x,1)=0, & 0<x<1;\\ u_k^0(0,y)=0, & u_k^0(1,y)=0, & 0<y<1;\\ v_k^0(x,0)=\varphi_k^0(x), & v_k^0(x,1)=0, & 0<x<1;\\ v_k^0(0,y)=\varphi_k^0(-y), & v_k^0(1,y)=0, & 0<y<1.\end{array} \tag{42}$$

To each of the latter sequences we apply the Gram–Schmidt orthogonalization procedure to obtain the sequences $\{u_k\}$ and $\{v_k\}$ orthogonal in $H_{harm}(D)$. Finally, we define on $\Gamma_0 = \{(x,0)\in\Omega\}\cup\{(0,y)\in\Omega\}$ the sequence $\{\varphi_k^x, \varphi_k^y, \varphi_k^{xy}\}$:

$$\varphi_k^x(x,y) = \begin{cases} u_k(-x,0), & -1<x<0,\ y=0,\\ 0, & \text{otherwise},\end{cases}$$

$$\varphi_k^y(x,y) = \begin{cases} u_k(-y,0), & x=0,\ -1<y<0,\\ 0, & \text{otherwise},\end{cases} \tag{43}$$

$$\varphi_k^{xy}(x,y) = \begin{cases} v_k(x,0), & 0<x<1,\ y=0,\\ v_k(0,y), & x=0,\ 0<y<1,\end{cases}$$

and

$$\begin{aligned} \varphi_k^{xy}(-x,0) &= v_k(x,0), \quad 0<x<1,\\ \varphi_k^{xy}(0,-y) &= v_k(0,y), \quad 0<y<1,\end{aligned} \tag{44}$$

or

$$\begin{aligned} \varphi_k^{xy}(-x,0) &= \tfrac{1}{2}(v_k(x,0)+v_k(0,x)), \quad 0<x<1,\\ \varphi_k^{xy}(0,-y) &= \tfrac{1}{2}(v_k(y,0)+v_k(0,y)), \quad 0<y<1,\end{aligned} \tag{45}$$

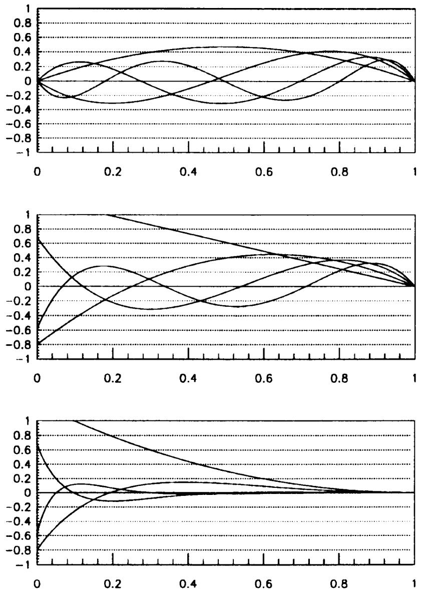

FIG. 1. *"Sample" basis functions.*

or else

$$\varphi_k^{xy}(-x,0) = v_k(x,x), \quad 0 < x < 1, \tag{46}$$
$$\varphi_k^{xy}(0,-y) = v_k(y,y), \quad 0 < y < 1.$$

Three plots on Fig. 1 display, from top to bottom, $\{\varphi_k^x(x,0),\ k = 1,\dots,4\}$, $\{\varphi_{2k}^{xy}(x,0),$ $\{k = 1,\dots,4\}$, and $\{\varphi_{2k}^{xy}(-x,0),\ k = 1,\dots,4\}$ (the latter for the case (46)) for $0 < x < 1$.

It has been shown in [21] that the sequence of sets $\Phi^n = \{\varphi_k^x, \varphi_k^y\}_{k=1,n-1}$, $\{\varphi_k^{xy}\}_{k=1,2n-1}$ is well conditioned in $H_0^{1/2}(\Gamma)$, therefore the PDM with the interface basis Φ^n converges at a rate independent of the dimension of the discrete problem. Furthermore, in cases (44) and (45) this basis is piecewise polynomial and therefore, according to Theorem 1, provides a high accuracy of the PDM approximations. In the case (46), where φ_k^{xy} are not polynomial on $\{(x,y) \in \Gamma : x < 0 \text{ or } y < 0\}$, one can use polynomial interpolations for $v_k(x,x)$ to preserve high accuracy. Our numerical experiments presented below in section 5 show not only a very mild dependence of the condition number of the PDM–Galerkin system (30) obtained with this type of basis on the number of basis functions but also a rather weak dependence of this condition number on the number of subdomains. Thus, in Tables 1–3 we can read the number of PDM iterations needed to solve the model problem $-\Delta u = f$ in $\Omega = (0,1)^2$ with Dirichlet boundary condition on $\partial\Omega$, with n basis functions on each side of Γ, and m subdomains in each direction (x and y).

We clearly observe that the basis constructed using (46) is much better conditioned than those using (44) and (45). For this reason, in our numerical tests presented in section 5 only the former basis was used.

Further theoretical investigation of the properties of such well-conditioned bases, in particular, the dependence of the condition number of the system (30) on the number of subdomains, is the subject of our forthcoming research.

Table 1
PDM iterations for basis (44).

m	4	6	8	10	12	14
n						
4	46	86	121	153	186	217
8	57	98	133	167	203	238
12	62	100	142	178	210	245
16	63	106	146	185	213	247

Table 2
PDM iterations for basis (45).

m	4	6	8	10	12	14
n						
4	45	75	102	129	156	184
8	63	107	141	179	226	274
12	69	117	164	214	262	313
16	71	118	166	219	275	317

Table 3
PDM iterations for basis (46).

m	4	6	8	10	12	14
n						
4	29	35	39	42	46	51
8	31	35	39	42	46	51
12	31	35	39	42	46	51
16	31	35	39	42	46	51

Consider now a general conformal rectangular decomposition. We denote $\Omega_i = A_iB_iC_iD_i$, where the sides A_iB_i and D_iC_i are parallel to the x-axis, and the sides A_iD_i and B_iC_i are parallel to the y-axis. It is evident that in order to define basis functions on the interface Γ, it is enough to define them on each pair $A_iB_i,\ A_iD_i$, and we do it according to the following algorithm. If $A_iB_i \in \Gamma$, while $A_iD_i \in \partial\Omega$, then we define on A_iB_i a set of basis functions, obtained from the traces of u_k on $\Gamma^x = [0,1] \times \{0\}$ through linear mapping $\Gamma^x \to A_iB_i$. The case $A_iB_i \in \partial\Omega,\ A_iD_i \in \Gamma$ is treated in a similar way. Finally, if $A_iB_i \in \Gamma,\ A_iD_i \in \Gamma$, then the subdomain $\Omega_i \equiv A_iB_iC_iD_i$ has the left-hand neighbor Ω_l and the bottom neighbor Ω_b, and to define basis functions on $D_iA_iB_i$ and their prolongations onto A_lA_i and A_iA_b (needed for the continuity) we use the functions φ_k^{xy} and the piecewise-linear mapping $[0,1]\times\{0\} \to A_iB_i,\ \{0\}\times[0,1] \to A_iD_i,\ [-1,0]\times\{0\} \to A_lA_i,\ \{0\}\times[-1,0] \to A_bA_i$. (If Ω_i has the right-hand and/or the top neighbor, then corresponding prolongations on A_iB_i and A_iD_i are to be defined as well.)

In the multidimensional case the construction scheme remains essentially the same. We start with a set of polynomials orthonormal in $H_{harm}(D)$, where D is the m-dimensional unit cube (they can be constructed, e.g., with the help of the *discrete separated variables* technique by solving 1D eigenvalue problems; see, e.g., [6]). Using prolongations analogues to (44), (45), or (46) we obtain a well-conditioned system of functions on $\Gamma_0 = \{(x_1, x_2, \ldots, x_m) \in (-1,1)^m : x_1 = 0 \text{ or } x_2 = 0 \text{ or } \ldots \text{ or } x_m = 0\}$. Finally, through corresponding mappings of Γ_0 onto the interface Γ, analogues to those used in the above algorithm, we construct a well-conditioned basis on Γ.

4. The spectral subdomain solvers. We have pointed out in section 3.1 that m Dirichlet problems must be solved in order to compute u_0. Then, at each iteration of CG method, m additional Dirichlet problems (see (35)) must be solved prior to the computation of the terms in (34). Each Dirichlet problem is set on a subdomain, and it is solved by the SCM illustrated above (see (18)). Below we make use of a subindex N to denote the approximants obtained by the application of the SCM in each subdomain. So u_0 is approximated by u_{0_N}, p^l by $p^l_{_N}$, and v^n by $v^n_{_N}$. We have for $k = 1, \dots, m$

$$\left\{ \begin{aligned} -\Delta u_{0_N} + \kappa u_{0_N} &= f \quad \text{at any } (x_i^{(k)}, y_j^{(k)}) \in \Omega_k, \\ u_{0_N} &= g \quad \text{at any } (x_i^{(k)}, y_j^{(k)}) \in \partial\Omega_k, \end{aligned} \right. \tag{47}$$

where $(x_i^{(k)}, y_j^{(k)})$ are the images of (x_i, y_j) under the mapping $\Phi^{(k)} : \Omega^* \to \Omega_k$, and for $l = 1, \dots,$ until convergence of CG

$$\left\{ \begin{aligned} -\Delta p^l_{_N} + \kappa p^l_{_N} &= 0 \quad \text{at any } (x_i^{(k)}, y_j^{(k)}) \in \Omega_k, \\ p^l_{_N} &= 0 \quad \text{at any } (x_i^{(k)}, y_j^{(k)}) \in \partial\Omega_k \backslash \Gamma, \\ p^l_{_N} &= p^l_{_\Gamma} \quad \text{at any } (x_i^{(k)}, y_j^{(k)}) \in \partial\Omega_k \cap \Gamma. \end{aligned} \right. \tag{48}$$

A function like $p^l_{_N}$ satisfying the differential equation $-\Delta p^l_{_N} + \kappa p^l_{_N} = 0$ at all collocation nodes internal to Ω_k below are referred to as *"discrete harmonic"* from now on.

4.1. Poincaré–Steklov operator in the spectral framework. As noticed in section 3.1 the linear system (30) can be solved by a CG method. This entails that at each CG iteration we have to apply the Poincaré–Steklov operator to the spectral collocation approximation of functions belonging to span$\{\Phi^n\}$.

To this aim we need to compute the coefficients q_i^l in (34), i.e.,

$$q_i^l = \sum_{k=1}^{m} \int_{\partial\Omega_k \cap \Gamma} \varphi_i^n \frac{\partial p^l}{\partial n_k} \, d(\partial\Omega)_k, \tag{49}$$

where the *interface basis functions* $\{\varphi_i^n\}$, defined on Γ, are polynomials of degree n on each segment of Γ (see Theorem 1). In SCM we substitute each integral on $\partial\Omega_k$ appearing in (49) by its discrete counterpart $(\cdot,\cdot)_{N,\partial\Omega_k}$ introduced above, i.e., (49) by

$$\sum_{k=1}^{m} \Big(\varphi_i^n, \frac{\partial p^l_{_N}}{\partial n_k} \Big)_{N,\partial\Omega_k}. \tag{50}$$

The degree N that we take must be not less than the degree n of the polynomial basis functions at the interfaces introduced in section 3.3. Moreover, the function $p^l_{_N}$ is a "discrete harmonic" polynomial satisfying a collocation problem of the form (48). Therefore, owing to (16), formula (50) reads

$$\sum_{k=1}^{m} a^{(k)}_{_N} (p^l_{_N}, w_i^n), \tag{51}$$

where $a^{(k)}_{_N}$ stands for the restriction to Ω_k of the discrete bilinear form $a_{_N}$ (15) and $w_i^n = E_\Gamma \varphi_i^n$ are the "harmonic" extensions to Ω of the interface basis functions φ_i^n (see section 3). The terms (51) can be computed effectively avoiding an explicit knowledge of the functions w_i^n. Indeed, let us consider on each subdomain Ω_k the set of the Lagrange polynomials $\{\chi_{_N}^{k,ij} \in \mathbb{P}_{_N}(\Omega_k)$ for $i,j = 0,\ldots,N\}$, such that $\chi_{_N}^{k,ij}(x_i^{(k)}, y_j^{(k)}) = 1$ while $\chi_{_N}^{k,ij}$ vanishes at all remaining collocation nodes of $\overline{\Omega}_k$. Then $w^n_{i\,|\Omega_k}$ admits the representation

$$w_i^n = \sum_{j_1=1}^{N} \sum_{j_2=1}^{N} \Phi_{j_1 j_2}^{i,n} \chi_{_N}^{k,j_1 j_2}, \tag{52}$$

where $\Phi_{j_1 j_2}^{i,n} = w_i^n(x_{j_1}^{(k)}, y_{j_2}^{(k)})$. Equation (51) becomes

$$\sum_{k=1}^{m} \sum_{j_1=1}^{N} \sum_{j_2=1}^{N} \Phi_{j_1 j_2}^{i,n} a_{_N}^{(k)}(p_{_N}^l, \chi_{_N}^{k,j_1 j_2}). \tag{53}$$

Since $p_{_N}^l$ is "discrete harmonic" in Ω_k, (53) becomes

$$\sum_{k=1}^{m} \sum_{j_1,j_2} \Phi_{j_1 j_2}^{i,n} a_{_N}^{(k)}(p_{_N}^l, \chi_{_N}^{k,j_1 j_2}), \tag{54}$$

where the sum on j_1 and j_2 is restricted only to the nodes of $\partial\Omega_k$. As a matter of fact, using (16), we have

$$a_{_N}^{(k)}(p_{_N}^l, \chi_{_N}^{k,j_1 j_2}) = -\Big(\frac{\partial p_{_N}^l}{\partial n_{_k}}, \chi_{_N}^{k,j_1 j_2}\Big)_{N,\partial\Omega_k} + (Lp_{_N}^l, \chi_{_N}^{k,j_1 j_2})_{N,\Omega_k},$$

where the latter term is equal to zero because $Lp_{_N}^l = 0$ at all internal nodes, while $\chi_{_N}^{k,j_1 j_2}$ are equal to zero at all boundary nodes. On the other hand, the first term on the right-hand side vanishes for all internal $j_1\ j_2$ as $\chi_{_N}^{k,j_1 j_2} = 0$ there.

Proceeding similarly with the use of the discrete integral on $\partial\Omega_k$ as in (50), we obtain for the coefficients $b(w_i^n)$ of (33) the following formula:

$$b(w_i^n) = -\sum_{k=1}^{m} \Big(\varphi_i^n, \frac{\partial u_{0_N}}{\partial n_k}\Big)_{N,\partial\Omega_k}. \tag{55}$$

For (47) we now obtain

$$b(w_i^n) = -\sum_{k=1}^{m} \Big[a_{_N}^{(k)}(u_{0_N}, w_i^n) - (f, w_i^n)_{N,\Omega_k}\Big] \tag{56}$$

and, from (52),

$$b(w_i^n) = -\sum_{k=1}^{m} \sum_{j_1=1}^{N} \sum_{j_2=1}^{N} \Phi_{j_1 j_2}^{i,n} \Big[a_{_N}^{(k)}(u_{0_N}, \chi_{_N}^{k,j_1 j_2}) - (f, \chi_{_N}^{k,j_1 j_2})_{N,\Omega_k}\Big]. \tag{57}$$

Finally, in view of (47), the coefficients (55) simplify to

$$b(w_i^n) = -\sum_{k=1}^{m} \sum_{j_1,j_2} \Phi_{j_1 j_2}^{i,n} \Big[a_{_N}^{(k)}(u_{0_N}, \chi_{_N}^{k,j_1 j_2}) - (f, \chi_{_N}^{k,j_1 j_2})_{N,\Omega_k}\Big], \tag{58}$$

where again the sum on j_1 and j_2 is restricted only to the nodes of $\partial\Omega_k$.

4.2. Remark on the solution of the algebraic problem. Consider the algebraic linear system associated to the approximation of the Dirichlet problems (47) or (48) in each subdomain:

$$L_{sp}\mathbf{v} = \mathbf{f}. \tag{59}$$

Here L_{sp} is the pseudospectral matrix associated to the differential operator: it is a squared matrix of dimension $\mathbf{N}{=}(N_x{+}1)\cdot(N_y{+}1)$ and it contains entries corresponding to the equations associated to both internal and boundary nodes; $\mathbf{v}$ is the vector of unknowns (internal and boundary) of dimension $\mathbf{N}$; $\mathbf{f}$ is the vector of the right-hand side of dimension $\mathbf{N}$ and it includes the boundary data. The matrix L_{sp} has a diagonal structure for the rows associated to the boundary nodes and a block sparse structure otherwise. Unfortunately, it is not symmetric. To overcome this drawback we multiply the rows of L_{sp} and $\mathbf{f}$ by the diagonal matrix W of quadrature weights ω_{ij}, eliminate the rows and the columns of the matrix associated to the boundary nodes, update the right-hand side $\mathbf{f}$, and compress the remaining elements of the matrix. In this way the dimension of the matrix is reduced from $\mathbf{N}$ to $\mathbf{N}_1 = (N_x - 1) \cdot (N_y - 1)$ and, after a suitable row permutation, the linear system (59) becomes

$$I_b\mathbf{v}_b = \mathbf{g}_b, \tag{60}$$

$$W_{int}(L_{sp})_{int}\mathbf{v}_{int} = W_{int}\mathbf{f}_{int} - (L_{sp})_b\mathbf{v}_b, \tag{61}$$

where $(L_{sp})_{int}$ (respectively, $(L_{sp})_b$) is the block of the matrix L_{sp} corresponding to the internal (respectively, boundary) nodes of the computational domain. The same notational convention is used for vectors $\mathbf{v}$, $\mathbf{f}$, and W, while I_b is the identity matrix acting on the boundary nodes. Now $\tilde{L} = W_{int}(L_{sp})_{int}$ is a symmetric and positive definite matrix.

The linear system (61) is solved by Cholesky factorization and subsequent forward and backward resolution. The matrices $\tilde{L}$ associated to subdomains are independent from the boundary conditions and so they can be assembled and factored once at Step 1 of the PD–SC algorithm (the computational cost is $\mathcal{O}(\mathbf{N}_1^3/6)$ of floating point operations for each subdomain), while at each CG iteration of Step 2 only the resolution is carried out (with a cost of $\mathcal{O}(\mathbf{N}_1^2)$ floating point operations for each subdomain).

4.3. The algorithm for the PD–SC method and its parallelism. Below we present a step-by-step description of the implementation of the PD–SC method using the notations introduced in the previous sections.

```
Initialization
Step0 Construction of basis functions.
Step1 Evaluation of the solution $u_{0_N}$ on the domains $\Omega_k$ and evaluation of the right-
        hand side $\hat{b}{=}b(w_i^n)$ for the linear system (30).
        The following algorithm is considered:
        for $k = 1, \ldots, m$
        a Dirichlet problem of the form (47) is solved on $\Omega_k$, the normal derivative
        of $u_{0_N}$ on the boundary of $\Omega_k$ is computed and the coefficients $b(w_i^n)$ of (32)
        are evaluated (see (58)).
Step2 Evaluation of $v_N^n$ by solving the linear system $\hat{S}\hat{a} = \hat{b}$ (30) by CG iterations:
Initialization
  $\hat{p}^0 = \hat{r}^0 = \hat{S}\hat{a}^0 - \hat{b}$:
            for $k = 1, \ldots, m$
        the computation of $\hat{r}^0 = \hat{S}\hat{a}^0 - \hat{b}$ is carried out on $\Omega_k$; i.e., a Dirichlet
        "harmonic" problem similar to (48) on $\Omega_k$ is solved by spectral collo-
        cation method in order to find the function $r^0_{_N}$ on $\Omega_k$, then the terms
        $a(r^0_{_N}, w^n_i)$ are computed as described in (50)–(54).
    $\mathrm{res}^0 = e = (\hat{r}^0, \hat{r}^0)$
    $l = 1, \ldots,$ *until convergence*
        $\hat{q}^l = \hat{S}\hat{p}^l$:
            ***for k = 1, . . . , m***
            the computation of $\hat{q}^l = \hat{S}\hat{p}^l$ is carried out on $\Omega_k$; i.e., a Dirichlet
            harmonic problem similar to (48) on $\Omega_k$ is solved by spectral col-
            location method in order to find the function $p^l_{_N}$ on $\Omega_k$, then the
            terms $a(p^l_{_N}, w^n_i)$ are computed as described in (50)–(54).
        $\alpha = (\hat{r}^l, \hat{r}^l)/(\hat{q}^l, \hat{p}^l)$
        $\hat{a}^l = \hat{a}^{l-1} - \alpha\hat{p}^l$
        $\hat{r}^l = \hat{r}^{l-1} - \alpha\hat{q}$
        $\mathrm{res}^l = (\hat{r}^l, \hat{r}^l)/e$
        if ($\mathrm{res}^l < \varepsilon$) convergence is obtained
        $\beta = \mathrm{res}^l/\mathrm{res}^{l-1}$
        $\hat{p}^l = \hat{r}^l + \beta\hat{p}^{l-1}$
**Step3** $u_{_N} = u_{0_N} + v^n_{_N}$ on all $\Omega$.
```

The algorithm offers a good level of parallelism. In fact, each time that the Poincaré–Steklov operator is invoked, the computations in the subdomains can run concurrently and the degree of parallelism of the algorithm is equal to the number of subdomains.

4.4. Computational complexity comparison. In this section we make a comparison between PDM and the other two domain decomposition methods—Dirichlet–Neumann method and spectral element method with additive Schwarz preconditioner—in terms of computational complexity. In particular, we focus our attention on the core of each algorithm, that is, the resolution of the CG iterations for PDM and additive Schwarz preconditioner method and the Richardson iteration for the Dirichlet–Neumann method. Since all three methods are based on spectral approximation, it is obvious that the computational cost in order to assemble the spectral matrices is the same for all of them. We note that the third method (CG iteration on the spectral element matrix preconditioned by the additive Schwarz method) requires the construction and the factorization of the finite element preconditioner matrices.

We recall that N is the polynomial degree in each direction, $\mathbf{N} = (N+1)^2$ is the number of nodes in each domain, $\mathbf{N}_1 = (N-1)^2$ is the number of rows and columns of the symmetrized matrices (see section 4.1), and M is the number of subdomains in the decomposition.

PD–SC method. Inside each CG iteration the most expensive part of the PDM is the computation of the product $\hat{q}^l = \hat{S}\hat{p}^l$. As we show in section 4.3, the computation of this product amounts to the resolution of a linear system arising from the spectral approximation of a Helmholtz problem with Dirichlet boundary conditions in each subdomain, and then the computation of the normal derivative on the interface. The local matrices (one per subdomain) are symmetric and positive definite and a Cholesky factorization can be used. The local matrices are assembled and decomposed (at $\mathcal{O}(\mathbf{N}_1^3/6)$ flops) when the solution u_0 is computed, so that inside each CG iteration we need to perform

(i) $\mathcal{O}\left(M \cdot \mathbf{N}_1^2\right)$ flops for the resolution of the linear systems;

(ii) $\mathcal{O}(M \cdot N^2)$ flops for the computation of the pseudospectral derivative on the interface; and

(iii) $\mathcal{O}(M \cdot \mathbf{N})$ flops for the computation of the coefficients of $\hat{q}^l$ (see formula (34)).

Dirichlet–Neumann algorithm. At each iteration of the Dirichlet–Neumann algorithm we have to solve $M/2$ Helmholtz problems with Dirichlet boundary conditions; therefore we have to compute the normal derivative across the interface and to solve $M/2$ Helmholtz problems with mixed boundary conditions. Finally, the relaxation parameter is computed by making use of the numerical solution on the interface. As in PDM, the local matrices are assembled and decomposed (by Cholesky factorization) on the first step of the procedure. The computational cost of each iteration can be summarized as follows:

(i) $\mathcal{O}\left(M \cdot \mathbf{N}_1^2\right)$ flops for the resolution of the linear systems;

(ii) $\mathcal{O}(M \cdot N^2)$ flops for the computation of the pseudospectral derivative on the interface; and

(iii) $\mathcal{O}(M \cdot N)$ flops for the computation of the relaxation parameter.

CG iteration on the spectral element matrix preconditioned by additive Schwarz algorithm with coarse grid. At each CG iteration we need to compute M local matrix–vector products with the spectral element matrices to solve M linear systems on the extended subdomains and to solve a linear system on the coarse grid. The computational cost can be summarized as follows:

(i) $\mathcal{O}(M \cdot \mathbf{N}^2)$ flops for the computation of the matrix–vector product;

(ii) $\mathcal{O}(M \cdot \mathbf{N}_2^2)$ flops for the resolution of the linear systems on the preconditioners, $\mathbf{N}_2$ is the number of nodes of the enlarged element, for a minimum overlap (one point) it is at most $\mathbf{N}_2 = (N+2)^2$; and

(iii) $\mathcal{O}(M^2)$ flops for the resolution of the linear system on the coarse mesh.

Remark. We observe that the computational cost per iteration for both the PDM and Dirichlet–Neumann method is similar, so below we make a CPU time comparison between PDM and the additive Schwarz preconditioner on the spectral element matrix.

5. Numerical results. Here we illustrate numerically some properties of the PD–SC method applied to problem (7)–(9) with inhomogeneous boundary data and various choices of the computational domain, the number of subdomains, and the degree of the spectral approximation.

We start with a systematic numerical comparison with the Dirichlet–Neumann algorithm (see Appendix A for a short description of this method). The right-hand side f and the boundary data g in (7)–(9) are chosen according to known analytical solutions. Unless otherwise specified, Dirichlet boundary conditions on the whole boundary are imposed. We denote by

$$err := \log_{10} \frac{||u_{N} - u||_{H^1(\Omega)}}{||u||_{H^1(\Omega)}} \tag{62}$$

and by NIT the number of CG iterations of the PDM or the number of iterations of the Dirichlet–Neumann scheme in order to achieve the solution of the problem with the tolerance $\varepsilon = 10^{-12}$.

First, we analyze numerically the behavior of NIT versus the number of basis functions N (or, equivalently, the polynomial degree of the spectral solution) on each

Table 4

Number of iterations and logarithm of the error. Ω *is split into* M *equal subdomains* $\Omega_i = (x_{a_i}, x_{b_i}) \times (y_a, y_b)$ *with* $N_x = N_y = N$ *and* κ =0. *The test solutions are* (a) $u(x,y) = \sin(7\pi x)\sin(7\pi y)$ *on* $\Omega = (0,4) \times (0,1)$ *partitioned into six aligned subdomains, and* (b) $u(x,y) = 1 - e^{\lambda x}\cos(2\pi y)$ *with* $\lambda = 20 - 2\sqrt{100+\pi^2}$ *on* $\Omega = (-1,1) \times (0,1)$ *partitioned into eight aligned subdomains.*

	PDM				Dirichlet–Neumann			
	solution (a)		solution (b)		solution (a)		solution (b)	
N	*err*	*NIT*	*err*	*NIT*	*err*	*NIT*	*err*	*NIT*
4	0.93	3	−1.12	19	0.93	26	−1.12	44
8	−0.51	7	−4.01	21	−0.52	22	−4.01	40
12	−1.64	9	−7.60	21	−1.65	16	−7.59	39
16	−3.29	9	−11.65	21	−3.29	7	−11.67	40
20	−5.42	10	−12.70	22	−5.42	4	−12.69	38
24	−7.91	10	−12.48	22	−7.91	4	−12.41	43
28	−10.69	10	−11.73	22	−10.70	4	−11.79	45
32	−13.02	10	−11.50	22	−13.02	4	−11.63	43

Table 5

Number of iterations and logarithm of the error. $\Omega = (0,2) \times (0,2)$ *is split into four equal subdomains having one internal crosspoint, with* $N = N_x = N_y$ *and* $\kappa = 0$. *The test solutions are* (a) $u(x,y) = \sin(7\pi x)\sin(7\pi y) + 1$, *and* (b) $u(x,y) = 1 - e^{\lambda x}\cos(5\pi y)$ *with* $\lambda = -10$.

	PDM				Dirichlet–Neumann			
	solution (a)		solution (b)		solution (a)		solution (b)	
N	*err*	*NIT*	*err*	*NIT*	*err*	*NIT*	*err*	*NIT*
4	1.64	7	−0.21	13	1.64	22	−0.21	28
8	−0.19	14	−1.20	19	−0.19	39	−1.21	36
12	−1.38	16	−3.22	21	−1.38	43	−3.22	46
16	−3.14	17	−5.71	21	−3.14	58	−5.71	62
20	−5.27	17	−8.60	22	−5.27	44	−8.60	57
24	−7.76	18	−11.80	22	−7.76	57	−11.80	68
28	−10.55	18	−12.63	22	−10.55	57	−11.57	66
32	−13.19	18	−12.49	23	−13.33	65	−12.50	62

Table 6

Number of iterations and logarithm of the error. $\Omega = (0,2) \times (0,2)$ *is split into* 16 *equal subdomains having nine internal crosspoints, with* $= N_x = N_y = N$ *and* $\kappa = 0$. *The test solutions are* (a) $u(x,y) = \sin(7\pi x)\sin(7\pi y) + 1$, *and* (b) $u(x,y) = 1 - e^{\lambda x}\cos(5\pi y)$ *with* $\lambda = -10$.

	PDM				Dirichlet–Neumann			
	solution (a)		solution (b)		solution (a)		solution (b)	
N	*err*	*NIT*	*err*	*NIT*	*err*	*NIT*	*err*	*NIT*
4	−0.12	29	−1.01	31	−0.12	53	−1.01	57
8	−2.16	31	−3.48	32	−2.16	74	−3.48	83
12	−4.71	32	−6.65	32	−4.71	86	−6.65	84
16	−7.78	32	−10.32	32	−7.78	89	−10.32	90
20	−11.24	32	−12.36	32	−11.24	102	−12.14	99
24	−12.37	32	−12.44	32	−12.81	102	−11.99	103

interface of the decomposition. The domain Ω is decomposed into M subdomains along one direction (stripwise partition) (see Table 4) or into $M \times M$ equal square subdomains (partition with internal crosspoints) (see Tables 5, 6, and 7).

In accordance with the theory, NIT is uniformly bounded with respect to N for both PDM and Dirichlet–Neumann procedures since the subdomain partition is free of crosspoints. Note that the convergence behavior of the Dirichlet–Neumann iterations is not monotone. For a fixed number of subdomains sharing internal crosspoints

TABLE 7

PDM iterations and logarithm of the error. $\Omega = (0,2) \times (0,2)$ *is split into* 16 *equal subdomains with* $N = N_x = N_y$ *and* $\kappa = 0$. *Neumann condition is imposed on vertical sides while Dirichlet condition is assigned to the horizontal ones. The test solutions are* (a) $u(x,y) = \sin(7\pi x)\sin(7\pi y)+1$, *and* (b) $u(x,y) = 1 - e^{\lambda x}\cos(5\pi y)$ *with* $\lambda = -10$.

	solution (a)		solution (b)	
N	err	NIT	err	NIT
4	1.72	28	0.27	29
8	−0.19	42	−1.04	41
12	−1.38	44	−3.12	42
16	−3.13	44	−5.64	44
20	−5.27	45	−8.54	44
24	−7.76	45	−11.75	46
28	−10.54	45	−12.39	45
32	−11.83	46	−12.24	45

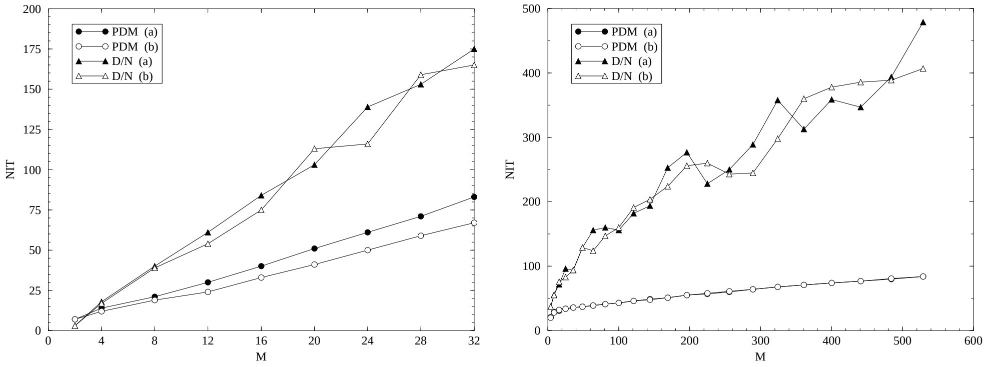

FIG. 2. *The number of iterations of PDM and Dirichlet–Neumann versus the number of subdomains.* Ω *is split into* M *equal subdomains* $\Omega_i = (x_{a_i}, x_{b_i}) \times (y_a, y_b)$ *without internal crosspoints* (*left*). $\Omega = (-1,2) \times (-1,2)$ *is split into* $M = M_1 \times M_1$ *equal subdomains with internal crosspoints* (*right*).

NIT depends logarithmically on the polynomial degree N for the Dirichlet–Neumann method [26], while it is uniformly bounded for the PDM method.

For a given domain Ω and a fixed number of unknowns in each subdomain, the rate of convergence depends on the number of subdomains M for both PDM and Dirichlet–Neumann procedures, as can be seen in Fig. 2 (left and right), the former referring to the stripwise partition of Ω, the latter to the crosswise partition. The test solutions are (a) $u(x,y) = 1 - e^{\lambda x}\cos(2\pi y)$ with $\lambda = 20 - 2\sqrt{100+\pi^2}$, and (b) $u(x,y) = \frac{y^2}{1+x^2}$. In both cases $N_x = N_y = 8$ and $\kappa = 0$. We observe that for both procedures the number of iterations depends rather mildly on the number of subdomains, but we have two different constants which control this behavior.

Now we turn to the comparison between the PDM and the method based on the additive Schwarz preconditioner for CG iterations on the spectral element matrix. We consider the problem (11) in the domain $\Omega^* = (0,1)^2$ with $\kappa = 0$; the test solution is $u(x,y) = e^{-\frac{1}{3}(x^3+y^3)+\frac{1}{2}(x^2+y^2)}$. In our experiments we used the code in which the amount of overlap is an input parameter. The results presented below refer to the minimum overlap, namely, the one involving only one gridpoint outside each subdomain because, according to all our numerical tests, this choice turned out to be the most effective one in terms of CPU time.

In Tables 8 and 9 we show the number of CG iterations executed until the convergence to the exact solution of the problem (11) with the tolerance $\varepsilon = 10^{-12}$. Table

Table 8

The number of CG iterations for the PDM and the additive Schwarz preconditioner with coarse grid solver for the problem (11) *with a stripwise decomposition of* Ω *into* M *subdomains. The tolerance is* $\varepsilon = 10^{-12}$.

PDM							
	M						
N	2	3	4	5	6	7	8
4	3	6	9	12	15	18	21
8	7	12	15	18	21	24	27
12	8	13	16	19	22	25	28
16	9	13	17	19	22	26	28
20	9	13	17	19	22	26	29

Additive Schwarz							
	M						
N	2	3	4	5	6	7	8
4	17	23	29	33	38	40	46
8	27	40	51	57	64	71	79
12	37	55	71	79	89	97	108
16	47	67	89	100	112	121	127
20	58	80	105	118	131	137	150

Table 9

The number of CG iterations for PDM and the additive Schwarz preconditioner with coarse grid solver for the problem (11) *with the decomposition of* Ω *into* $M = M_1 \times M_1$ *equal subdomains. The tolerance is* $\varepsilon = 10^{-12}$.

PDM							
	M_1						
N	2	3	4	5	6	7	8
4	7	21	30	33	35	37	38
8	14	28	32	33	35	37	39
12	17	28	32	34	35	37	39
16	17	29	32	34	35	37	39
20	17	30	33	34	35	37	39

Additive Schwarz							
	M_1						
N	2	3	4	5	6	7	8
4	19	32	36	39	39	39	40
8	28	35	38	40	39	40	40
12	38	49	51	55	53	55	55
16	49	61	64	69	71	71	71
20	60	68	75	80	79	79	80

Table 10

CG iteration number for solving the problem (11) *up to a tolerance* $\varepsilon = 10^{-5}$ *by the PDM. In the squared brackets the CG iteration numbers for additive Schwarz preconditioner with one point overlap are shown.*

N	M								
	2	3	4	5	6	7	8	9	10
3	5 [8]	12 [13]	14 [17]	15 [18]	16 [19]	17 [19]	18 [19]	20 [19]	21 [19]
4	7 [8]	12 [14]	14 [15]	15 [16]	16 [16]	17 [16]	19 [16]	19 [16]	21 [16]
5	9 [11]	13 [14]	14 [16]	15 [17]	16 [17]	17 [17]	19 [17]	19 [17]	21 [17]
6	8 [12]	13 [15]	14 [16]	15 [17]	16 [17]	17 [18]	19 [17]	20 [17]	21 [17]
7	9 [13]	13 [15]	14 [15]	15 [16]	16 [16]	17 [16]	19 [16]	20 [16]	21 [16]
8	9 [14]	13 [15]	14 [15]	15 [16]	16 [14]	17 [16]	19 [15]	20 [16]	21 [16]
9	9 [16]	13 [15]	14 [16]	15 [17]	16 [16]	17 [16]	19 [16]	20 [18]	21 [18]
12	9 [19]	13 [20]	14 [19]	15 [21]	16 [19]	17 [20]	19 [20]	20 [20]	21 [20]
16	10 [24]	13 [24]	14 [21]	15 [23]	16 [23]	17 [23]	19 [23]	20 [23]	21 [23]

8 corresponds to the stripwise partition of the computational domain into M subdomains and Table 9 corresponds to the partition into M subdomains with internal crosspoints.

We also present the number of CG iterations executed until the convergence to the exact solution of problem (11) up to a tolerance $\varepsilon = 10^{-5}$ (see Table 10). We observe that for this higher tolerance the additive Schwarz method in some cases performs better than PDM.

We have measured the CPU time required by each CG iteration for both the PDM and the additive Schwarz preconditioner on the same workstation (Risc6000 IBM Mod. 370). In Table 11 the total CPU times for the stripwise decomposition are shown (above for the PDM and below for the additive Schwarz preconditioner), while in Table 12 the total CPU times for the crosspoint decomposition are shown. In both tables the CPU times refer to the CG iterations reported in Tables 8 and 9 with toler-

Table 11

The total CPU time measured in milliseconds for the PDM (above) and additive Schwarz preconditioner (below). The data refer to the solution of the problem (11) *with the stripwise decomposition of Ω^* into M subdomains and to the CG iteration numbers reported in Table* 8.

	M						
N	2	3	4	5	6	7	8
4	3.04	5.75	9.88	15.00	21.54	29.36	38.26
	15.37	25.87	42.95	55.17	71.96	84.39	139.33
8	21.58	44.66	69.03	99.00	134.81	176.00	224.66
	104.66	186.60	289.71	392.11	508.54	647.06	800.49
12	95.60	185.07	304.24	398.78	540.54	678.13	852.87
	470.85	887.15	1404.97	1900.70	2507.95	3185.97	3936.45
16	319.36	555.02	859.95	1170.70	1534.45	1987.68	2417.18
	1546.69	2906.97	4716.72	6434.04	8543.80	10495.45	12618.07
20	911.55	1491.75	2191.89	2984.50	3881.92	4951.59	6301.68
	4227.66	7869.06	13474.92	17941.08	22779.07	27662.58	34139.42

Table 12

The total CPU time measured in milliseconds for the PDM (above) and the additive Schwarz preconditioner (below). The data refer to the decomposition of Ω^* into $M = M_1 \times M_1$ equal subdomains and to the CG iteration numbers reported in Table 9.

	M_1						
N	2	3	4	5	6	7	8
4	8.10	42.10	101.87	174.19	266.32	379.00	507.85
4	35.53	108.56	221.46	379.75	598.61	901.24	1273.57
8	65.03	254.56	500.94	798.74	1233.16	1713.09	2347.84
8	233.52	642.04	1229.71	2093.38	3164.05	4601.22	6493.65
12	288.87	939.52	1747.79	3043.03	4462.52	6381.32	8694.95
12	1055.09	2903.59	5503.28	9481.33	14069.77	19766.69	26831.16
16	863.34	2760.78	5201.39	8626.60	12716.75	17854.33	24110.91
16	3440.03	9874.47	17949.77	30869.14	45804.02	64031.11	87763.55
20	2186.28	6926.20	13079.41	20929.89	30569.26	43960.94	59317.23
20	9280.80	24029.37	51796.53	79068.67	116775.66	161249.29	212681.62

Table 13

The total CPU time measured in milliseconds for the PDM (above) and the additive Schwarz preconditioner (below). The data refer to the decomposition of Ω^* into $M = M_1 \times M_1$ equal subdomains and to the CG iteration numbers reported in Table 10.

	M_1						
N	2	3	4	5	6	7	8
4	8.10	26.17	51.95	85.99	131.42	187.20	269.59
4	29.37	80.48	154.89	256.93	392.99	573.72	783.01
8	47.98	138.91	257.94	420.74	661.26	897.09	1279.84
8	191.24	476.64	861.25	1427.14	2063.05	3014.82	4036.90
12	196.07	545.02	910.79	1656.03	2478.73	3524.72	4986.15
12	803.91	1951.81	3498.16	5866.79	8402.65	11765.34	15785.86
16	650.89	1674.38	3039.59	5047.00	7573.45	10845.23	14528.51
16	2524.28	6232.93	10593.33	17369.33	25670.42	35598.79	48392.51
20	1723.58	4350.19	7976.01	12921.01	19119.48	27408.34	37822.43
20	6464.58	14970.13	29469.68	43306.02	64612.53	87938.63	113352.40

ance $\varepsilon = 10^{-12}$. The CPU times of Table 13 refer instead to the calculations reported in Table 10 with relative tolerance $\varepsilon = 10^{-5}$. We note that the construction of the "sample" basis functions (43) used for generating the well-conditioned basis on Γ does not depend on the data of the problem. This task can therefore be accomplished independently of the problem at hand, and should not be regarded as an initialization step.

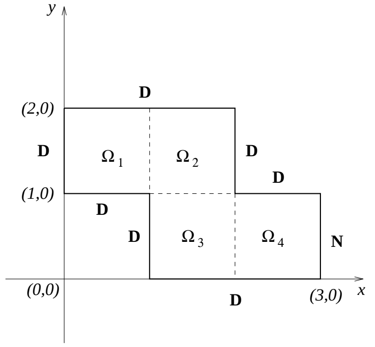

FIG. 3. *The computational domain of the first example. The capitals D and N refer to the Dirichlet and Neumann conditions, respectively.*

TABLE 14
The results on the first test case.

	$M = 4$		$M = 16$		$M = 64$	
N	*err*	*NIT*	*err*	*NIT*	*err*	*NIT*
4	−0.40	12	−1.78	39	−3.12	43
8	−2.67	23	−5.28	48	−7.81	42
12	−5.73	24	−9.30	49	−12.41	43
16	−9.16	25	−12.17	52	−12.41	43

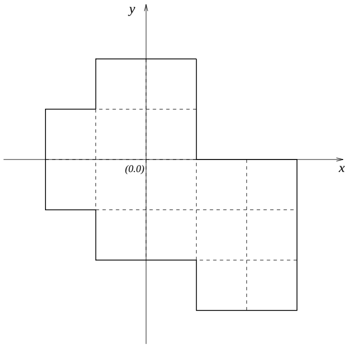

FIG. 4. *The computational domain of the second example. A Dirichlet condition is imposed on the boundary.*

We now consider a couple of examples that refer to two computational domains of more complex form. Both problems are solved by the PD–SC method.

In the first example we consider the domain of Fig. 3. and a decomposition in 4, 16, and 64 subdomains. The first subdivision is shown in Fig. 3, while the other subdivisions are obtained by splitting each subdomain into four equal subdomains. The test solution is $u(x, y) = 1 - e^{\lambda x} \cos(3\pi y)$ with $\lambda = -10$, $\kappa = 0$, and $N = N_x = N_y$. We impose a Dirichlet boundary condition on the whole boundary except on $\partial\Omega^N = \{(x, y) : 0 \leq y \leq 1, x = 3\}$. In Table 14 we present the logarithm of the relative error in the $H^1(\Omega)$ norm and the number of CG iterations executed until the convergence up to the relative tolerance $\varepsilon = 10^{-12}$.

In the second example we consider the domain of Fig. 4 and the decomposition into 16, 64, and 256 subdomains. The first subdivision is shown in Fig. 4, while the other

Table 15
The results on the second test case.

	$M = 16$		$M = 64$		$M = 256$	
N	err	NIT	err	NIT	err	NIT
4	-0.45	31	-1.80	40	-3.12	61
8	-2.68	33	-5.29	40	-7.81	61
12	-5.79	34	-9.31	40	-12.35	62
16	-9.16	34	-12.07	41	-12.35	62

subdivisions are obtained by splitting each subdomain into four equal subdomains. The test solution is $u(x, y) = 1-e^{\lambda x}\cos(3\pi y)$ with $\lambda = -10$, $\kappa = 1$, and $N = N_x = N_y$. In Table 15 we present the logarithm of the error in $H^1(\Omega)$ norm and the number of CG iterations executed until the convergence up to the relative tolerance $\varepsilon = 10^{-12}$.

As noted above, the number of iterations depends on M and is uniformly bounded with respect to N.

Appendix A. The Dirichlet–Neumann domain decomposition algorithm. In order to give a short description of the Dirichlet–Neumann algorithm we consider again the decomposition (19), introducing the following notations to represent interfaces between subdomains:

$$\Gamma_{ij} := \partial\Omega_i \cap \partial\Omega_j \text{ for } i, j = 1, \ldots, m, \quad \Gamma := \bigcup_{i,j=1}^{m} \Gamma_{ij}. \tag{63}$$

We denote by u_i the restriction of u to the subdomain Ω_i for $i = 1, \ldots, m$. The differential problem (1) can be written in the equivalent multidomain formulation (see [26]) $\forall i = 1, \ldots, M$

$$\begin{cases} Lu_i \equiv -\Delta u_i + \kappa u_i = f & \text{in } \Omega_i, \\ u_i = 0 & \text{on } \partial\Omega_i \cap \partial\Omega, \\ u_i = u_j & \text{on } \Gamma_{ij}, \\ \frac{\partial u_i}{\partial n_i} = -\frac{\partial u_j}{\partial n_j} & \text{on } \Gamma_{ij}. \end{cases} \tag{64}$$

The last two equations of (64) are the *transmission conditions* between subdomains Ω_i and Ω_j; the first one ensures the continuity of the solution across the interfaces, while the second condition stresses the continuity of the flux.

We consider the domain Ω to be split in black and white subdomains, like a chessboard, and we associate the condition of continuity of the solution to the black subdomains and the continuity of the flux to the white ones. Then we start an iterative procedure to solve the differential Dirichlet subproblems on black subdomains and, subsequently, mixed Dirichlet–Neumann problems on white subdomains. In order to ensure the convergence of the iterative scheme to the solution of the original problem, the relaxation of the solution is needed at each iteration and it is chosen dynamically according to an optimality criterion (see [26], [13]).

Appendix B. The additive Schwarz preconditioner on the spectral element approximation. The problem (1) can be approximated by the conformal spectral element method based on the GLL quadrature formulas (see [18]). Every domain Ω_k of the nonoverlapping decomposition (19) is considered as a spectral element, and interpolation of degree N (in both x and y directions) is used on it. The conformal spectral element discretization produces a linear system that is solved by the

CG iterations, preconditioned by a modified version of the additive Schwarz preconditioner with coarse grid solver (see [10], [11]). We denote by $\hat{\Omega}_k$ (with $k = 1, \ldots, m$) the extension of the element Ω_k, internal to $\overline{\Omega}$, and we define the modified version of the additive Schwarz preconditioner P_{as}^{-1} as

$$P_{as}^{-1} = R_0^T A_0^{-1} R_0 + \sum_{k=1}^{Ne} R_{k,e}^T A_{k,e}^{-1} R_{k,e}. \tag{65}$$

Here $A_{k,e}$ is the local matrix generated by the bilinear finite element discretization of the original problem at the discretization nodes inside $\hat{\Omega}_k$, while $R_{k,e}$ is a restriction map from the nodes in Ω to the nodes in $\hat{\Omega}_k$, and $R_{k,e}^T$ is an extension map from $\hat{\Omega}_k$ to Ω (see [8]). The union of all the quadrature nodes on the elements Ω_k induces a fine mesh, while the vertices of the nonoverlapping spectral element Ω_k induce a coarse mesh. A_0 stands for the bilinear finite element matrix arising from the discretization of the original problem on the coarse mesh and R_0 stands for the restriction map from the coarse mesh to the global fine mesh. This preconditioner differs from that of Dryja and Widlund: the approximation used by the preconditioner is not the same as used for the discretization of the problem; nevertheless, the results obtained with this preconditioner are quite satisfactory (see [24], [15]).

Acknowledgments. We are indebted to Professor V. I. Agoshkov for fruitful and helpful advice.

REFERENCES

[1] V. Agoshkov, *Domain Decomposition Methods: Some Results of Theory and Applications*, in Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, Y. A. Kuznetsov, G. Meurant, J. Périaux, and O. Widlund, eds., SIAM, Philadelphia, PA, 1991, pp. 3–18.

[2] V. Agoshkov and V. Lebedev, *The Poincaré-Steklov's operators and the domain decomposition algorithms in variational problems*, Computational Processes and Systems, 2 (1985), pp. 173–227; Nauka, Moscow (in Russian).

[3] V. Agoshkov and E. Ovtchinnikov, *Projection decomposition method*, Math. Models Methods Appl. Sci., 4 (1994), pp. 773–794.

[4] C. Bernardi and Y. Maday, *Approximations Spectrales de Problèmes aux Limites Elliptiques*, Springer–Verlag, Paris, 1992.

[5] J. Bramble, J. Pasciak, and A. Schatz, *An iterative method for elliptic problems on regions partitioned into substructures*, Math. Comp., 46 (1986), pp. 361–369.

[6] C. Canuto and D. Funaro, *The Schwarz algorithm for spectral methods*, SIAM J. Numer. Anal., 25 (1988), pp. 24–40.

[7] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer–Verlag, Berlin, 1988.

[8] T. Chan and T. Mathew, *Domain decomposition algorithms*, in Acta Numerica, Cambridge University Press, London, 1994, pp. 61–143.

[9] P. Davis and P. Rabinowitz, *Methods of Numerical Integration*, Academic Press, London, New York, 1984.

[10] M. Dryja and O. Widlund, *Some domain decomposition algorithms for elliptic problems*, in Iterative Methods for Large Linear Systems, L. Hayes and D. Kincaid, eds., Academic Press, New York, 1989, pp. 273–291.

[11] M. Dryja and O. Widlund, *Additive Schwarz methods for elliptic finite element problems in three dimensions*, in Proc. Fifth Conf. on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, D. Keyes, G. Meurant, J. Scroggs, and R. Voigt, eds., SIAM, Philadelphia, PA, 1992.

[12] M. Dryja and O. Widlund, *Some recent results on Schwarz type domain decomposition algorithms*, in Proc. Sixth Conf. on Domain Decomposition Methods for Partial Differential Equations, A. Quarteroni et al., eds., American Mathematical Society, Providence, RI, 1994, pp. 53–62.

[13] D. Funaro, A. Quarteroni, and P. Zanolli, *An iterative procedure with interface relaxation for domain decomposition methods*, SIAM J. Numer. Anal., 25 (1988), pp. 1213–1236.

[14] E. Gagliardo, *Caratterizzazione delle tracce sulla frontiera relative ad alcune classi di funzioni in n variabili*, Rend. Sem. Mat. Univ. Padova, 27 (1957), pp. 284–305.

[15] P. Gervasio, *Risoluzione di equazioni alle derivate parziali con metodi spettrali in regioni partizionate in sottodomini*, Ph.D. thesis, Department of Mathematics, Università degli Studi di Milano, Milano, Italy, 1995.

[16] D. Gottlieb and S. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications*, SIAM, Philadelphia, PA, 1977.

[17] J. L. Lions and E. Magenes, *Nonhomogeneous Boundary Value Problems and Applications*, Springer–Verlag, Berlin, 1972.

[18] Y. Maday and A. Patera, *Spectral element methods for the incompressible Navier-Stokes equations*, in State-of-the-Art Surveys on Computational Mechanics, A. K. Noor and J. T. Oden, eds., ASME, New York, 1989.

[19] S. Mikhlin and K. L. Smolitskii, *Approximate Methods for Solving Differential and Integral Equations*, Nauka, Moscow, 1965 (in Russian).

[20] S. Orszag, *Spectral methods for problem in complex geometries*, J. Comput. Phys., 37 (1980), pp. 70–92.

[21] E. Ovtchinnikov, *The construction of a well-conditioned basis for the projection decomposition method*, Calcolo, 30 (1993), pp. 255–271.

[22] A. Patera, *A spectral element method for fluid dynamics: Laminar flow in a channel expansion*, J. Comput. Phys., 54 (1984), pp. 468–488.

[23] L. Pavarino, *Domain Decomposition Methods for the p-Version Finite Element Method for Elliptic Problems*, Ph.D. thesis, Department of Mathematics, Courant Institute of Mathematical Sciences, New York, 1992.

[24] S. Phal, *Schwarz Type Domain Decomposition Methods for Spectral Element Discretizations*, Ph.D. thesis, Department of Mathematics, University of the Witwatersand, South Africa, 1993.

[25] A. Quarteroni, *Domain decomposition and parallel processing for the numerical solution of partial differential equations*, Surveys Math. Indust., 1 (1991), pp. 75–118.

[26] A. Quarteroni and G. S. Landriani, *Domain decomposition preconditioners for the spectral collocation methods*, J. Sci. Comput., 3 (1988), pp. 45–75.

[27] A. Quarteroni and A. Valli, *Theory and application of Steklov-Poincaré operators for boundary-value problems*, in Applied and Industrial Mathematics, R. Spigler, ed., Kluwer Academic Publishers, Dordrecht, the Netherlands, 1990, pp. 179–203.

[28] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, Springer–Verlag, Heidelberg, 1994.

[29] G. Yakovlev, *On traces of functions from Sobolev spaces on piecewise-smooth surfaces*, Math. Coll., 74(4) (1967), pp. 526–542 (in Russian).

[30] P. Zanolli, *Domain decomposition algorithms for spectral methods*, Calcolo, 24 (1987), pp. 201–240.