8

An Iterative Substructuring Preconditioner for Collocation with Hermite Bicubics

Gabriel Mateescu & Calvin J. Ribbens[1]

Introduction

We propose and evaluate a novel Schur complement type preconditioner for solving linear systems of equations arising from piecewise Hermite bicubic collocation discretization of elliptic partial differential equations (PDEs) with mixed boundary conditions. Although collocation is a very general and effective discretization technique for many PDE problems, relatively little is known about combining domain decomposition techniques with collocation. In this paper we define a new preconditioner in terms of two special grids—a coarse grid and a hybrid coarse/fine grid—which together provide the framework for approximating the interface subproblem. We illustrate the performance of the preconditioner by combining it with a Krylov subspace method (KSM) and reporting experimental results on three test problems.

We consider the family of boundary value problems:

$$\mathbf{Lu} = \mathbf{g} \quad \text{in } \Omega \qquad \text{and} \qquad \mathbf{Bu} = \mathbf{t} \quad \text{on } \partial\Omega, \tag{1}$$

where $\mathbf{L}$ is a linear second-order elliptic operator, $\mathbf{B} = \mathbf{p}\,\frac{\partial}{\partial \mathbf{n}} + \mathbf{q}$, $\Omega = (0, a) \times (0, a)$ and $\partial\Omega$ is the boundary of Ω. The discretization method is collocation with piecewise

1 Authors address: Computer Science Department, Virginia Tech, Blacksburg, VA 24061 USA Email: {mateescu, ribbens}@vt.edu This work was supported in part by DARPA ARO grant DAAH04-94-G-0010 and by Virginia Tech Interdisciplinary Center for Applied Mathematics.
Eleventh International Conference on Domain Decomposition Methods
Editors Choi-Hong Lai, Petter E. Bjørstad, Mark Cross and Olof B. Widlund

Hermite bicubic basis functions; for convenience, we refer to this method as *Hermite collocation* (HC). Hermite collocation provides $O(h^4)$ accuracy, where h is the fine grid step, for sufficiently smooth problems, and does not require the computation of any integrals for generating the coefficient matrix. Let $A\mathbf{u} = \mathbf{f}$ be the discrete problem induced by applying HC to (1), where $A \in \mathbf{R}^{\mathbf{N}\times\mathbf{N}}$. The solution method has two elements: (i) a right preconditioner M which is employed to obtain the preconditioned system $AM^{-1}\mathbf{v} = \mathbf{f}$, where $\mathbf{v} = \mathbf{Mu}$; (ii) a KSM, such as the Generalized Minimal Residual (GMRES) method [SS86], for solving the preconditioned system. The matrix A induced by HC is not symmetric even when $\mathbf{L}$ is the Laplacian; moreover, A is typically ill-conditioned and unpreconditioned GMRES takes $O(N)$ iterations.

Domain decomposition based on Schur complement methods (SCMs) can provide robust preconditioners for HC problems. SCMs work by partitioning the domain in *subdomains* and an *interface* consisting of *edges* and *vertices*, and deriving a reduced Schur complement problem for the interface subproblem (see [SBG96]). However, constructing SCM preconditioners for HC is not straightforward. Well known strategies such as *tangential preconditioning* [CK90], *Neumann-Dirichlet* [BW86],or *multilevel diagonal scaling* [BPX90] do not generalize easily to the HC problem derived from (1). Previous work on domain decomposition methods for HC problems has been limited to symmetric positive definite (SPD) problems and problems obtained by symmetrizing the initial problem. Bialecki, *et al.* have proposed several preconditioners for solving Poisson's equation with HC. One method [Bia93] is based on a decomposition in parallel strips of the domain and a SCM preconditioner. Another method [BCDF94] employs the additive Schwarz approach. Our preconditioner is remarkable in that it supports general elliptic operators, two-dimensional decomposition (important for scalability), and proposes a new method for approximately solving the Schur complement problem.

The Hermite Collocation Problem

We proposed in [MRW98] a preconditioner based on a strip decomposition of the domain. This paper generalizes that work to a preconditioner for substructuring in two dimensions, which we call the *edge-vertex preconditioner* (EVP).

Let n_0 be an integer greater than one, and $X = \{x_k\}_{k=0}^{n_0-1}$ and $Y = \{y_m\}_{m=0}^{n_0-1}$ be partitions of $[0, a]$, where $x_k = k\,h$, $y_m = m\,h$, and $h = a/(n_0 - 1)$. The set of grid points, or *nodes*, in $\bar{\Omega} = \Omega \cup \partial\Omega$ is $G^h = X \times Y$. In HC, the unknown function u is approximated by a function U in the space $\mathcal{V}$ of continuously differentiable piecewise bicubic polynomials. Let $n_G = n_0^2$ be the number of nodes in G^h. Following [HMR85], with each node n in G^h there are associated four Hermite bicubic basis functions, Φ_n^i, $1 \le i \le 4$, centered at n and with support $[x_k - h, x_k + h] \times [y_m - h, y_m + h]$, where (x_k, y_m) are the coordinates of n. The set $\{\Phi_n^i \mid 1 \le i \le 4, 1 \le n \le n_G\}$ is a basis for $\mathcal{V}$, and U can be written as $U(x, y) = \sum_{i=1}^{4} \sum_{n=1}^{n_G} U_n^{(i)} \Phi_n^i(x, y)$, where $U_n^{(1)}$, $U_n^{(2)}$, $U_n^{(3)}$, and $U_n^{(4)}$ are the values U, U_y, U_x, and U_{xy} at n. The HC problem is to find $U \in \mathcal{V}$ which satisfies (1) on a set $\mathcal{G} \cup \partial\mathcal{G}$ of $N = 4n_G$ *collocation points*. A good choice for $\mathcal{G}$ and $\partial\mathcal{G}$ uses the tensor product of two Gauss points per interval (see [HMR85]).

Schur Complement Preconditioning

Let $D = \{1, 2, \ldots, N\}$ denote the index set of all unknowns (equations). Substructuring performs a symmetric permutation of A such that subdomain indices (I) are first, then edge indices (E), then vertex indices (V). This induces a partition $D = I \cup E \cup V$. Let $B = E \cup V$ be the set of interface indices. The unknown vector $\mathbf{u}$ can be written as $\mathbf{u} = (\mathbf{u_I}, \mathbf{u_B})^{\mathbf{T}}$, where $\mathbf{u_B} = (\mathbf{u_E}, \mathbf{u_V})^{\mathbf{T}}$. For simplicity, assume a perfectly uniform partition, where every subdomain and edge has N_I and N_E unknowns, respectively. If N_V is the number of vertex unknowns, then $N = n_I N_I + n_E N_E + N_V$, where n_I and n_E are the number of subdomains and edges respectively.

The operator A after substructuring is

$$A = \begin{pmatrix} A_{II} & A_{IB} \\ A_{BI} & A_{BB} \end{pmatrix}, \text{with } A_{BB} = \begin{pmatrix} A_{EE} & A_{EV} \\ A_{VE} & A_{VV} \end{pmatrix}, A_{II} = \sum_{i=1}^{n_I} R_{I_i}^T A_{II}^{(i)} R_{I_i}. \quad (2)$$

The first (second) subscript of a block denotes the index set of the equations (unknowns) in that block, and $A_{BI} = (A_{EI}\, A_{VI})^T$, $A_{IB} = (A_{IE}\, A_{IV}) \neq A_{BI}^T$ even when $\mathbf{L}$ is SPD. The above structure of A is called *arrowhead*, since A_{II} is block diagonal with n_I blocks. Let $S = A_{BB} - A_{BI} A_{II}^{-1} A_{IB}$ be the Schur complement with respect to A_{BB}. A block triangular factorization of A^{-1} is

$$A^{-1} = \begin{pmatrix} I & -A_{II}^{-1} A_{IB} \\ 0 & I \end{pmatrix} \begin{pmatrix} A_{II}^{-1} & 0 \\ 0 & S^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -A_{BI} A_{II}^{-1} & I \end{pmatrix}. \quad (3)$$

A preconditioner for A employs a preconditioner $\hat{S}$ to S in (3), since S is dense and expensive to construct. Moreover, in general, $\hat{S}^{-1}$ is not explicitly computed, but its action on a vector $\mathbf{v_B}$ is computed instead. The seminal paper [BPS86] introduced the Bramble-Pasciak-Schatz (BPS) preconditioning strategy which uses a *coarse grid operator* A_H to provide global coupling between subdomains, and an *interpolation map* $R_H^T : V \mapsto B$ to approximate the solution on B. The BPS preconditioner is $\hat{S}_{BPS}^{-1} = \sum_{i=1}^{n_E} R_i^T S_{i,i}^{-1} R_i + R_H^T A_H^{-1} R_H$, where $R_i : B \mapsto E_i$ is the pointwise restriction map from B to the indices of the ith edge, and $S_{i,i}$ is the Schur complement for the ith edge. Note that the BPS method does not specify a scheme for approximating $S_{i,i}^{-1}$.

The Preconditioner

Substructuring applied to the HC problem induces the linear system $A\,\mathbf{u} = \mathbf{f}$, where $\mathbf{u} \in \mathbf{R^N}$, and which can be written as three subproblems:

$$A_{II}\mathbf{u_I} + \mathbf{A_{IE}u_E} + \mathbf{A_{IV}u_V} = \mathbf{f_I} \text{ (subdomain subproblem)} \quad (4)$$

$$A_{EI}\mathbf{u_I} + \mathbf{A_{EE}u_E} + \mathbf{A_{EV}u_V} = \mathbf{f_E} \text{ (edge subproblem)} \quad (5)$$

$$A_{VI}\mathbf{u_I} + \mathbf{A_{VE}u_E} + \mathbf{A_{VV}u_V} = \mathbf{f_V} \text{ (vertex subproblem)} \quad (6)$$

In (5) $A_{EE} = A_{EE}^e + A_{EE}^c$, with A_{EE}^e block diagonal, where a block comprises the degrees of freedom for an edge, and the superscripts e and c are abbreviations for "eigen" (own) and "coupled", respectively.

Our preconditioner is defined in terms of three grids: (i) the fine grid, on which the solution is sought; (ii) a hybrid fine/coarse grid, called the *edge grid*, associated with each edge, on which an approximate solution to (5) is found; and (iii) a *coarse grid* on which an approximate solution to (6) is found. The discretization on each grid is performed using the bicubics associated with that grid, e.g., the coarse grid basis functions centered at an interior vertex (x_k, y_m) have support $[x_k - H - h, x_k + H + h] \times [y_m - H - h, y_m + H + h]$, where H is the coarse grid parameter (see next paragraph). The collocation points are determined in terms of the fine grid only. Therefore, the right hand sides f_E and f_V are preserved when collocating on the edge and coarse grids, but the coefficient matrices change. Slightly abusing the notation, we denote by $(\mathbf{u_I}, \mathbf{u_E}, \mathbf{u_V})^{\mathbf{T}}$ the solution to the problem (4–6) as well as the solution to the approximate problem obtained using the three-grid discretization outlined above.

For simplicity, assume the domain Ω is decomposed into $n_I = (n_0 - 1)^2$ subdomains. To achieve perfect load balancing, suppose also that the fine grid is defined so that each subdomain contains exactly K^2 interior nodes. Figure 1 illustrates the case $\sqrt{n_I} = K = 3$. Let $H = Kh$ be the diameter of the corner subdomains. Subdomains away from $\partial\Omega$ have diameter $H + h$, while those along one side of $\partial\Omega$ are of size $(H + h) \times H$ or $H \times (H + h)$. Note also that nodes on $\partial\Omega$ are either vertex or interior nodes in this terminology.

We illustrate the definition of the edge grid by considering a vertical edge E_i. Assume E_i does not intersect $\partial\Omega$ (horizontal edges, and edges intersecting $\partial\Omega$ are handled similarly). Suppose E_i has endpoints (x, y) and $(x, y + H + h)$. The edge grid G_i associated with E_i is defined as $G_i = \{x - H, x, x + H\} \times \{y, y + h, y + 2h, \ldots, y + H + h\}$. Figure 1 shows the edge grid associated with E_4. Let V^i be the space of piecewise bicubic polynomials on $\mathcal{E}_i = [x - H, x + H] \times [y, y + H + h]$, where the Hermite bicubic basis functions have support $[-H, H] \times [-h, h]$ with respect to the nodes where they are centered. We approximate the restriction of u to $\mathcal{E}_i$ by an element in V^i. The collocation equation is $\hat{U}(\xi, \zeta) = \sum_{l=1}^{4} \sum_{n=1}^{4} \hat{U}_{k_n}^{(l)} \hat{\Phi}_{k_n}^{l}(\xi, \zeta)$ where $(\xi, \zeta) \in \mathcal{G} \cup \partial\mathcal{G}$ and $\hat{\Phi}_{k_n}^{l}$, $1 \le l, n \le 4$ are basis functions for V^i centered at the four nodes k_n closest to (ξ, ζ). Hence, we obtain the blocks $\hat{A}_{EE}$, $\hat{A}_{EI}$, and $\hat{A}_{EV}$ with elements $\mathbf{L}\hat{\mathbf{\Phi}}_{\mathbf{k_n}}^{\mathbf{l}}$ and $\mathbf{B}\hat{\mathbf{\Phi}}_{\mathbf{k_n}}^{\mathbf{l}}$, instead of A_{EE}, A_{EI}, and A_{EV} whose elements are $\mathbf{L}\mathbf{\Phi}_{\mathbf{k_n}}^{\mathbf{l}}$ and $\mathbf{B}\mathbf{\Phi}_{\mathbf{k_n}}^{\mathbf{l}}$. The matrix form of the edge problem is

$$\hat{A}_{EI}\mathbf{u_I} + \hat{\mathbf{A}}_{\mathbf{EE}}\mathbf{u_E} = \mathbf{f_E} - \hat{\mathbf{A}}_{\mathbf{EV}}\mathbf{u_V}, \text{ where } \hat{\mathbf{A}}_{\mathbf{EE}} = \hat{\mathbf{A}}_{\mathbf{EE}}^{\mathbf{e}} + \hat{\mathbf{A}}_{\mathbf{EE}}^{\mathbf{c}}, \tag{7}$$

with $\hat{A}_{EE}^{e}$ block diagonal. Similarly, we collocate on the coarse grid, approximate u in the function space corresponding to the coarse grid, and get the approximate vertex problem

$$\hat{A}_{VI}\mathbf{u_I} + \hat{\mathbf{A}}_{\mathbf{VV}}\mathbf{u_V} = \mathbf{f_V}. \tag{8}$$

The EVP provides an approximate solution to the system (4, 7, 8). We use implicit preconditioning, i.e., we compute the action of M^{-1} on a given vector $\mathbf{v}$ by solving the system $M\,\mathbf{u} = \mathbf{v}$. To construct the EVP we have to solve two problems: (i) decouple the coarse grid problem from the subdomains; (ii) find an approximation of (7) which removes the mutual edge coupling among the n_E edge subproblems due to $\hat{A}_{EE}^{c}$, so that the edge problems can be solved independently.

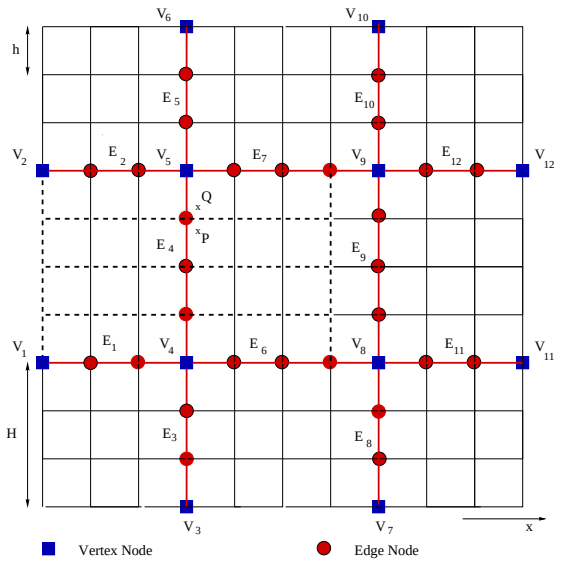

Figure 1 Three-level grid. The coarse grid is shown in thick solid lines. The edge grid for edge E_4 is shown in dashed lines. Edge nodes are shown as disks and vertex nodes as squares. P and Q are two of the collocation points associated with edge E_4.

The first problem is solved by observing that the vertex-subdomain coupling in (8) is due to the four corners of Ω which are subdomain nodes. Let $\bar{\mathbf{u}}_V = (\mathbf{u_C}, \mathbf{u_V})^{\mathbf{T}}$, where $\mathbf{u_C}$ is associated with the four corners. Let $\bar{A}_{VV}$ be the coarse grid operator obtained by expanding $\hat{A}_{VV}$ with coarse grid collocation at the collocation points associated with the corners. We get the independent vertex problem $\bar{A}_{VV}\,\bar{\mathbf{u}}_V = \bar{\mathbf{v}}_V$, where $\bar{\mathbf{v}}_V = (\mathbf{v_C}, \mathbf{v_V})^{\mathbf{T}}$.

Given $\mathbf{u_V}$, we solve the second problem by evaluating the approximate solution defined by $\mathbf{u_V}$ at those edge nodes that cause mutual edge coupling. Notice that only the edge nodes at a distance h from a vertex node cause mutual coupling. For example, the edge E_4 in Figure 1 is coupled to E_6 and E_7 by way of the nodes $(2H, H)$ and $(2H, 2H + h)$, respectively. Let $\tilde{\mathbf{u}}_E$ be the approximate value of the edge unknowns obtained by evaluating the piecewise bicubics defined by $\mathbf{u_V}$:

$\tilde{\mathbf{u}}_E = I_V^E \mathbf{u_V}$, where I_V^E denotes Hermite cubic interpolation from V to E unknowns.

We decouple the n_E edge problems by making the approximation $\hat{A}^c_{EE}\mathbf{u_E} \approx \hat{\mathbf{A}}^{\mathbf{c}}_{\mathbf{EE}}\tilde{\mathbf{u}}_{\mathbf{E}}$ which gives $\hat{A}_{EE}\mathbf{u_E} \approx \hat{\mathbf{A}}^{\mathbf{e}}_{\mathbf{EE}}\mathbf{u_E} + \hat{\mathbf{A}}^{\mathbf{c}}_{\mathbf{EE}}\tilde{\mathbf{u}}_{\mathbf{E}}$ and (7) becomes

$$\hat{A}_{EI}\mathbf{u_I} + \hat{\mathbf{A}}^{\mathbf{e}}_{\mathbf{EE}}\mathbf{u_E} = \mathbf{v_E} - \hat{\mathbf{A}}^{\mathbf{c}}_{\mathbf{EE}}\tilde{\mathbf{u}}_{\mathbf{E}} - \hat{\mathbf{A}}_{\mathbf{EV}}\mathbf{u_V}. \tag{9}$$

The action of M^{-1} is computed in two main stages. First we solve for $\bar{\mathbf{u}}_V$ directly, $\bar{\mathbf{u}}_V = \bar{A}_{VV}^{-1}\bar{\mathbf{v}}_V$. Second we solve approximately the system consisting of (9) and the

1 $\bar{\mathbf{u}}_V = \bar{A}_{VV}^{-1}\,\bar{\mathbf{v}}_V$;	▷ Solve coarse grid problem
2 $\tilde{\mathbf{u}}_E = I_V^E \mathbf{u_V}$;	▷ Approximate edge unknowns
3 $\tilde{\mathbf{u}}_I = A_{II}^{-1}\,(\mathbf{v_I} - \mathbf{A_{IE}}\tilde{\mathbf{u}}_\mathbf{E} - \mathbf{A_{IV}u_V})$;	▷ Approximately solve subdomains
4 $\mathbf{v_E} = \mathbf{v_E} - \hat{\mathbf{A}}_\mathbf{EI}\tilde{\mathbf{u}}_\mathbf{I} - \hat{\mathbf{A}}_\mathbf{EE}^\mathbf{c}\tilde{\mathbf{u}}_\mathbf{E} - \hat{\mathbf{A}}_\mathbf{EV}\mathbf{u_V}$;	▷ New right-hand side for interface
5 $\mathbf{u_E} = \hat{\mathbf{A}}_\mathbf{EE}^{\mathbf{e}\,-\mathbf{1}}\mathbf{v_E}$;	▷ Solve interface
6 $\mathbf{u_I} = \mathbf{A_{II}^{-1}}(\mathbf{v_I} - \mathbf{A_{IE}}\,\mathbf{u_E} - \mathbf{A_{IV}}\,\mathbf{u_V})$;	▷ Solve subdomains

Figure 2 Edge-Vertex Preconditioning Algorithm: $\mathbf{u} = \mathbf{M^{-1}}\,\mathbf{v}$

subdomain problem (see (4)): $A_{II}\mathbf{u_I} + \mathbf{A_{IE}u_E} = \mathbf{v_I} - \mathbf{A_{IV}u_V}$. This is done in three steps. First, we solve approximately the subdomain problem by replacing $\mathbf{u_E}$ with $\tilde{\mathbf{u}}_E$: $A_{II}\tilde{\mathbf{u}}_I = \mathbf{v_I} - \mathbf{A_{IE}}\tilde{\mathbf{u}}_\mathbf{E} - \mathbf{A_{IV}u_V}$, for $\tilde{\mathbf{u}}_I$. Second, we plug $\tilde{\mathbf{u}}_I$ in (9) which we approximate by $\hat{A}_{EE}^e\mathbf{u_E} = \mathbf{v_E} - \hat{\mathbf{A}}_\mathbf{EI}\tilde{\mathbf{u}}_\mathbf{I} - \hat{\mathbf{A}}_\mathbf{EE}^\mathbf{c}\tilde{\mathbf{u}}_\mathbf{E} - \hat{\mathbf{A}}_\mathbf{EV}\mathbf{u_V}$. This is equivalent to approximating $\hat{A}_{EE}^e\mathbf{u_E} - \hat{\mathbf{A}}_\mathbf{EI}\mathbf{A_{II}^{-1}A_{IE}}(\mathbf{u_E} - \tilde{\mathbf{u}}_\mathbf{E}) \approx \hat{\mathbf{A}}_\mathbf{EE}^\mathbf{e}\mathbf{u_E}$. This approximation is justified by the relation (see [Mat98] pages 53–54, for a proof):

$$\|\hat{A}_{EI}A_{II}^{-1}A_{IE}\ \mathbf{u}\|_\infty = \mathcal{O}\left(\frac{\mathbf{h}}{\mathbf{H}}\right)\ (\mathbf{1} + \|\mathbf{A_{EI}A_{II}^{-1}A_{IE}}\|_\infty)\ \|\mathbf{u}\|_\infty,\ \mathbf{u} \in \mathbf{R}^{\mathbf{n_E N_E}}.$$

Third, we solve for $\mathbf{u_I}$: $A_{II}\mathbf{u_I} = \mathbf{v_I} - \mathbf{A_{IE}u_E} - \mathbf{A_{IV}u_V}$. Figure 2 gives the steps to apply M^{-1} to $\mathbf{v} \in \mathbf{R^N}$.

The floating point operation cost of applying EVP is $C_P \approx 16NK + 8\frac{N^{1.5}}{K^3} + 5N + 58\frac{N}{K}$. The best choice for K is—if we account for the cost C_F of block factorizations—$K = O(N^{1/6})$, which gives $C_P = O(N^{7/6})$, $C_F = O(N^{4/3})$. The algorithm has good parallelism, except for step 1. To avoid a sequential bottleneck, the size of the coarse problem should be kept small enough.

Numerical Experiments

We employ GMRES with restart parameter 30, and the stopping rule is relative residual reduction of $\varepsilon = 10^{-5}$. The code is written in C and Fortran with parallel directives, and uses LAPACK's general band factorization and triangular solve procedures. The CPU time is given in seconds and is measured using the C-library function `times`. The tests are run on an SGI Origin 2000 machine with 16 processors running at 195-MHz. We use the following model problems defined on $\Omega = (0,2)\times(0,2)$ (drawn from the examples in Appendix A of [RB85]).

Problem 1. Problem 1 in [RB85] has a self-adjoint operator:

$$\left(e^{xy}\ u_x\right)_x + \left(e^{-xy}\ u_y\right)_y - (1+x+y)^{-1}u = g_1, \tag{10}$$

where subscripts indicate partial differentiation and with g_1 such that $u_1 = \frac{3}{4}e^{xy}\sin(\pi x)\sin(\pi y)$ is the true solution. The boundary conditions are Dirichlet on

one side and Neumann on the other three sides: $u = 0$ on $x = 2$ and $\frac{\partial u}{\partial \mathbf{n}} = \frac{\partial u_1}{\partial \mathbf{n}}$ on $y = 0$, $y = 2$, and $x = 0$.

Problem 2. Problem 2 in [RB85] has a general operator:

$$u_{xx} + (1 + y^2)\, u_{yy} - u_x - (1 + y^2)\, u_y = g_2, \tag{11}$$

with g_2 such that $u_2 = e^{x+y} + (x^2 - x)^2 \log(1 + y^2)$ is the solution. The boundary conditions are again Dirichlet on one side and Neumann on the other three: $u = u_2$ on $x = 2$ and $\frac{\partial u}{\partial \mathbf{n}} = \frac{\partial u_2}{\partial \mathbf{n}}$ on $y = 0$, $y = 2$, and $x = 0$.

Problem 3. Problem 12 in [RB85] has oscillatory coefficients of u_x and u:

$$u_{xx} + u_{yy} + (1 + \sin(\alpha x))\, u_x - \cos(\alpha y)\, u = g_3, \tag{12}$$

with g_3 determined by the solution $u_3 = \cos(\beta y) + \sin\beta(x - y)$, and with Dirichlet boundary conditions $u = u_3$ on $\partial\Omega$. We set $\alpha = \beta = \pi$.

Table 1 summarizes the performance of EVP on the three test problems. For constant problem size per subdomain (i.e., $K = H/h$ constant) the number of iterations is essentially constant, independent of H and h; for Problems 2 and 3 the number of iterations is reduced as N grows. For constant N, increasing the number of subdomains n_I can improve convergence as well. For example, for Problems 2 and 3 with $N = 36100$, increasing n_I from 6^2 to 8^2 reduces the number of iterations by one. This improvement is due to increasing coarse problem size; but eventually the reduction in iteration count is offset by the reduced efficiency from the sequential bottleneck of the coarse solve.

Table 2 compares EVP with PETSc's Additive Schwarz preconditioner [BGMSsc] with exact subdomain solves, for Problem 2 and constant N. Note that the ASM preconditioner is algebraic, i.e., it requires no knowledge of the PDE. We set the number of ASM blocks to p. EVP clearly outperforms ASM for this case. It is possible that ASM's performance would improve with a reordering of equations and unknowns.

Table 1 EVP performance.

K	n_I	N	Problem 1			Problem 2			Problem 3		
			Its	T_p	Mfl/sp	Its	T_p	Mfl/sp	Its	T_p	Mfl/sp
11	6^2	20164	17	1.6	29	5	0.7	35	5	0.7	37
	8^2	36100	21	2.8	27	3	0.8	32	4	0.9	34
	12^2	81796	21	5.3	21	2	1.5	25	3	1.6	26
	16^2	145924	20	11.4	13	1	2.0	25	2	2.7	19
15	6^2	36100	19	3.6	35	4	1.4	45	5	1.6	42
	8^2	64516	22	6.2	30	3	1.8	43	4	2.1	40
	10^2	101124	23	9.2	27	2	2.4	38	3	2.7	37
	12^2	145924	23	12.	24	2	3.1	36	3	3.5	35

Table 2 EVP versus Additive Schwarz. Problem 2, $N = 158404$, δ is the overlap, $S = T_1/T_p$ is the speedup, Its is the iteration count.

Processors	EVP, $n_I = 100, K = 19$				Additive Schwarz, $\delta = 2$			
	Its	Mflop/s per proc	T_p (sec)	S	Its	Mflop/s per proc	T_p (sec)	S
1	2	82	26.9	1	1	18	5053	1
2	2	80	13.8	1.9	19	16	1757	2.9
4	2	67	8.19	3.3	30	12	1257	4.0
8	2	55	5.01	5.4	67	19	490	10.3
10	2	50	4.39	6.1	92	13	606	8.3
12	2	51	3.63	7.4	117	11	642	7.9
14	2	48	3.29	8.2	120	9	736	6.9
16	2	45	3.08	8.7	150	7	831	6.1

REFERENCES

[BCDF94] Bialecki B., Cai X.-C., Dryja M., and Fairweather G. (1994) An additive Schwarz algorithm for piecewise Hermite bicubic orthogonal spline collocation. In *Domain Decomposition Methods in Science and Engineering*, volume 157 of *Contemporary Mathematics*, pages 237–244. AMS, Providence, RI.

[BGMSsc] Balay S., Gropp W., McInnes L. C., and Smith B. (1998, http://www.mcs.anl.gov/petsc) Petsc 2.0 User's Manual. Report ANL 95/11 - Revision 2.0.22, Argonne National Laboratory, Argonne, IL.

[Bia93] Bialecki B. (April 1993) A fast domain decomposition Poisson solver on a rectangle for Hermite bicubic orthogonal spline collocation. *SIAM J. Numer. Anal.* 30(2): 425–434.

[BPS86] Bramble J. H., Pasciak J. E., and Schatz A. H. (1986) The construction of preconditioners for elliptic problems by substructuring, I. *Math. Comp.* 47(175): 103–134.

[BPX90] Bramble J. H., Pasciak J. E., and Xu J. (1990) Parallel multilevel preconditioners. *Math. Comp.* 55: 1–22.

[BW86] Bjørstad P. E. and Widlund O. B. (1986) Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM J. Numer. Anal.* 23(6): 1093–1120.

[CK90] Chan T. F. and Keyes D. E. (1990) Interface preconditioning for domain-decomposed convection-diffusion operators. In Chan T., Glowinski R., Périaux J., and Widlund O. (eds) *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 245–262. SIAM, Philadelphia, PA.

[HMR85] Houstis E., Mitchell W., and Rice J. (December 1985) Collocation software for second-order elliptic partial differential equations. *ACM Trans. Math. Software* 11(4): 379–412.

[Mat98] Mateescu G. (1998) *Domain Decomposition Preconditioners for Hermite Collocation Problems.* PhD thesis, Virginia Tech, Blacksburg, VA.

[MRW98] Mateescu G., Ribbens C. J., and Watson L. T. (1998) A domain

decomposition algorithm for Hermite collocation problems. In *Proceedings of the Fifth Copper Mtn. Conf. on Iterative Methods*. Copper Mtn., CO.

[RB85] Rice J. R. and Boisvert R. F. (1985) *Solving Elliptic Problems Using ELLPACK*. Springer-Verlag, New York.

[SBG96] Smith B. F., Bjørstad P. E., and Gropp W. (1996) *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press.

[SS86] Saad Y. and Schultz M. H. (1986) GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comp.* 7: 856–869.