A Domain Decomposition Algorithm for Hermite Collocation Problems

Gabriel Mateescu, Calvin J. Ribbens, and Layne T. Watson *

Abstract

Designing substructuring methods for high order discretization methods for solving elliptic PDEs poses several challenges. We consider linear systems of equations arising from piecewise Hermite bicubic collocation applied to general linear two-dimensional second-order elliptic PDEs with mixed boundary conditions. We construct an efficient, parallel preconditioner for the GMRES(k) method. Our initial focus is on rectangular domains decomposed in one dimension. The main contribution of the paper is a novel interface preconditioner derived using the Schur complement approach. The Hermite collocation framework allows us to define a local discretization for the interface subproblems based on a hybrid fine-coarse mesh. Interface equations based on this mesh depend only weakly on unknowns associated with subdomains. This suggests an approximation to the Schur complement that is both accurate and efficient.

We describe our preconditioner and illustrate its performance with a numerical experiment. Our results indicate that for a fixed number of subdomains, the number of iterations is independent of the problem size; for fixed local problem size, the number of iterations grows very slowly with the number of subdomains.

1 Introduction

We are interested in designing parallel iterative solution methods applicable to high order discretization methods for linear second-order two-dimensional elliptic PDEs. Parallelism is essential for solving large problems. We describe a new preconditioning algorithm for a Krylov subspace iterative solution of linear systems arising from a piecewise Hermite bicubic collocation discretization of elliptic PDEs defined on a two-dimensional domain. For convenience, we refer to the collocation method as *Hermite collocation* (HC).

Hermite collocation provides $O(h^4)$ accuracy, where h is the grid step, for sufficiently smooth problems, making it an attractive alternative to linear finite elements when higher order accuracy is needed. Moreover, collocation methods do not require the computation of any integrals.

We employ the Generalized Minimal Residual (GMRES) method [SS86], whose strengths include generality and robustness. The discrete operator A generated by HC is ill-conditioned and unpreconditioned GMRES generally fails; basic preconditioners such as point Jacobi also fail. Therefore, finding a good preconditioner is critical for solving HC problems with a Krylov-subspace method.

The main contribution of the paper is a preconditioner for solving Hermite collocation problems with Krylov subspace methods. The preconditioner can be applied to a second-order elliptic PDE defined on a rectangular domain Ω and with general boundary conditions. In the framework of substructuring, we introduce a novel interface preconditioner, which we call the *edge preconditioner*. The edge preconditioner is constructed on a hybrid fine-coarse grid defined for each interface. The proposed method has high parallelism and early experimental results indicate good convergence and accuracy.

Bialecki and Dryja [BD93], Bialecki and Fairweather [BF97], and Bialecki [Bia97] have examined the applicability of multigrid, multilevel, and additive Schwarz methods for solving PDEs discretized with Hermite collocation. These authors proposed using the multilevel preconditioner of Bramble, Pasciak and Xu (BPX) [BPX90] and the square of the discrete Laplace operator. Furthermore, the authors have assumed Dirichlet boundary conditions. We are not aware of another preconditioner derived by substructuring for Hermite collocation problems with general boundary conditions.

*Department of Computer Science, Virginia Tech, Blacksburg, Virginia 24061, USA.

The remainder of the paper is organized as follows. After describing the HC discretization of a boundary value problem in Section 2, we present background information and related work in Section 3. The edge preconditioner is described in Section 4. Numerical results are presented in Section 5 followed by concluding remarks in Section 6.

2 Problem Formulation

2.1 The Hermite Collocation Problem

Let $\Omega = (0,1) \times (0,1)$ and $\partial\Omega$ its boundary. We consider the boundary value problem (BVP):

$$\mathbf{L}u = g \quad \text{in } \Omega \qquad \text{and} \qquad \mathbf{B}u = t \quad \text{on } \partial\Omega, \tag{1}$$

where $\mathbf{L}$ is a linear elliptic operator and $\mathbf{B} = p\,\frac{\partial}{\partial \mathbf{n}} + q$. Let n_0 be an integer greater than one, and define the partitions $X = \{x_k\}_{k=0}^{n_0-1}$ and $Y = \{y_m\}_{m=0}^{n_0-1}$ of $[0,1]$, where $x_k = k\,h$, $y_m = m\,h$, and $h = 1/(n_0 - 1)$. The set of grid points, or *nodes*, in $\bar{\Omega} = \Omega \cup \partial\Omega$ is $G^h = X \times Y$.

Collocation methods approximate the unknown function u by a function U in a finite dimensional function space V. In Hermite collocation, V is the space of continuously differentiable piecewise bicubic polynomials. Formally, $V = V_x \otimes V_y$, where $V_z = \{v \in C^1[0,1] : v|_{[z_k, z_{k+1}]} \in \mathbb{P}_3, k = 0, \ldots, n_0 - 2\}$, $z \in \{x, y\}$, and $\mathbb{P}_3$ is the set of polynomials of degree at most 3. Note that we consider general boundary conditions, so the functions in V are not necessarily zero on $\partial\Omega$.

Let $n_G = n_0^2$ be the number of nodes in G^h, and $\mathcal{N} : G^h \mapsto \{1, \ldots, n_G\}$ be a node ordering. Following [HMR85], with each node n in G^h, $n = \mathcal{N}(x_k, y_m)$, there are associated four *Hermite bicubic* functions, $\Phi_n^{(i)}$, $i = 1,2,3,4$, centered at n and with support $[x_{k-1}, x_{k+1}] \times [y_{m-1}, y_{m+1}]$. The set $\mathcal{B} = \{\Phi_n^{(i)} \mid 1 \le i \le 4, 1 \le n \le n_G\}$ is a *canonical basis* for V. The expression of a function $U \in V$ in the basis $\mathcal{B}$ is

$$U(x,y) = \sum_{i=1}^{4} \sum_{n=1}^{n_G} U_n^{(i)} \Phi_n^i(x,y), \tag{2}$$

where $U_n^{(1)} = U(x_k, y_m)$, $U_n^{(2)} = \frac{\partial U}{\partial y}(x_k, y_m)$, $U_n^{(3)} = \frac{\partial U}{\partial x}(x_k, y_m)$, $U_n^{(4)} = \frac{\partial U}{\partial xy}(x_k, y_m)$. Notice that we have implicitly defined a *type order* of the unknowns associated with a node; that is, the subscripts 1, 2, 3, 4 are associated with U, $\frac{\partial U}{\partial y}$, $\frac{\partial U}{\partial x}$, $\frac{\partial U}{\partial xy}$, respectively. Let $\mathcal{G}$ and $\partial\mathcal{G}$ be a set of *collocation points* in in Ω and on $\partial\Omega$, respectively, where the set $\mathcal{G} \cup \partial\mathcal{G}$ has $4n_G$ elements. A good strategy for $\mathcal{G}$ and $\partial\mathcal{G}$ selects the cross product of sets of Gauss points—two per interval in each direction. The Hermite collocation problem is to find $U \in V$ which satisfies (1) at the collocation points, i.e.,

$$\mathbf{L}U(\xi,\zeta) = g(\xi,\zeta),\ (\xi,\zeta) \in \mathcal{G}, \quad \text{and} \quad \mathbf{B}U(\xi,\zeta) = t(\xi,\zeta),\ (\xi,\zeta) \in \partial\mathcal{G},$$

where U is given by (2). The above relations induce a linear system $A\,\mathbf{u} = \mathbf{f}$, where $N = 4n_G$, $\mathbf{u} \in \mathbb{R}^N$ is the unknown vector, and $A \in \mathbb{R}^{N \times N}$.

2.2 Consistent Order

In order to be able to apply substructuring methods to the BVP, we need a *consistent order* of the equations in the linear system. We call an ordering consistent if it guarantees that the ith equation contains the ith unknown, for each $i, 1 \le i \le N$. The element order of the equations [DR84] which provides minimum bandwidth is not appropriate, because it is not consistent.

Assume that the nodes have been numbered in *natural order*, i.e., by scanning the nodes in south-to-north and west-to-east order and assigning consecutive indices to the nodes. Unknowns are naturally ordered by using the type order of unknowns associated with one node (described in Subsection 2.1) and following the node order.

We employ the consistent order proposed by Dyksen and Rice [DR84] who have described it as follows: *The grid intersection points are numbered in a natural way from west to east, south to north. The collocation points (equations) are then associated with the nearest grid point and are numbered in groups of four in the order of their corresponding grid point.*

3 Background and Related Work

3.1 Substructuring Methods

Substructuring works by partitioning the problem domain Ω into nonoverlapping *subdomains* and *artificial boundaries* or interfaces. For one dimensional partitioning, we obtain a *strip decomposition*, where the interface consists of *edges* between two subdomains. For two-dimensional partitioning, the interface consists of edges and *vertices* which lie on the interface of more than two subdomains.

Let Ω be partitioned into n_I subdomains, $\Omega_1, \Omega_2, \ldots, \Omega_{n_I}$, E (V) denote the set of edges (vertices), and n_E (n_V) denote the number of edges (vertices) induced by substructuring. Therefore, $\Omega = (\cup_{i=1}^{n_I} \Omega_i) \cup B$, where $B = E \cup V$, $E = \{E_1, \ldots, E_{n_E}\}$, and $V = \{V_1, \ldots, V_{n_V}\}$.

Notation: We denote the set of indices for unknowns associated with subdomain Ω_i, by I_i, and the set of all subdomain indices by $I = \cup_{i=1}^{n_I} I_i$. We use the symbol E_i to denote both the sets of points on the ith edge and the finite set of indices of the unknowns associated with the ith edge. Likewise, V indicates the set of vertex indices. After substructuring, the unknown vector $\mathbf{u}$ can be written as $\mathbf{u} = (\mathbf{u}_I, \mathbf{u}_B)^T$. In addition, we use superscripts to indicate subcomponents, e.g., $\mathbf{u}_I^{(i,i+1)}$ denotes the part of $\mathbf{u}_I$ associated with the subdomains Ω_i and Ω_{i+1}.

For simplicity, assume a perfectly even partition, i.e., every subdomain and edge has N_I and N_E unknowns, respectively. If N_V is the number of vertex unknowns, then $N = n_I N_I + n_E N_E + N_V$. Denote the set of unknown indices $\{1, 2, \ldots, N\}$ by D.

Applying substructuring reorders the nodes, unknowns and equations in a *substructuring order* (SO). In SO, the nodes in the subdomains Ω_i are numbered first, followed by the nodes that lie on edges, and vertex nodes are numbered last. The discrete operator A after SO has the *arrowhead* structure:

$$A = \begin{pmatrix} A_{II} & A_{IB} \\ A_{BI} & A_{BB} \end{pmatrix}, \tag{3}$$

where the first subscript denotes whether the equations in that matrix are subdomain (I) or interface (B) equations, while the second subscript denotes whether the coefficients are applied to subdomain (I) or interface (B) unknowns. Notice that, for HC discretization, $A_{BI} \neq A_{IB}^T$ even if $\mathbf{L}$ is symmetric positive definite.

It is customary to resort to restriction operators for denoting matrix blocks arising from substructuring. Define $R_I : D \mapsto I$ and $R_B : D \mapsto B$ to be the pointwise restrictions acting on the set of all indices D and selecting the subdomain and interface coefficients, respectively. Similarly, define the pointwise restrictions $R_{I_i} : I \mapsto I_i$, $R_{E_i} : E \mapsto E_i$, $R_j : B \mapsto E_j$, and $R_V : B \mapsto V$, where $i = 1, \ldots, n_I$, $j = 1, \ldots, n_E$. The transpose of a restriction acts as a prolongation map. The matrix $A_{II} = R_I A R_I^T$ is block diagonal, with a block $A_{II}^{(i)} = R_{I_i} A_{II} R_{I_i}^T$ for each subdomain. The *Schur complement* of A with respect to A_{II}

$$S = A_{BB} - A_{BI} A_{II}^{-1} A_{IB}, \tag{4}$$

allows us to write a block-LU factorization of A^{-1} as:

$$A^{-1} = \begin{pmatrix} I & -A_{II}^{-1} A_{IB} \\ 0 & I \end{pmatrix} \begin{pmatrix} A_{II}^{-1} & 0 \\ 0 & S^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -A_{BI} A_{II}^{-1} & I \end{pmatrix}. \tag{5}$$

3.2 Preconditioners Derived using Substructuring

We need to replace S^{-1} in (5) by a preconditioner for two reasons. First, S is a dense matrix and explicitly constructing it is expensive. Second, inverting a dense matrix is expensive as well.

Preconditioners are derived in the substructuring framework following a *two-scale procedure* [CM94]: there is a *fine grid*, with diameter h, and a *coarse* grid, with diameter H, associated with Ω. The coarse grid provides the global transfer of information which is needed because the solution $\mathbf{u}(\mathbf{x}_i)$ of an elliptic PDE at any point $\mathbf{x}_i$ depends on all the components of $\mathbf{f}$. The explicit dependence is expressed by a Green's function [Hac92].

A baseline preconditioner for S —proposed by Chan and Mathew [CM94]—approximates S by its diagonal blocks,

$$\hat{S}_{CM}^{-1} = \sum_{i=1}^{n_E} R_i^T S_{i,i}^{-1} R_i + R_V^T S_{VV}^{-1} R_V, \tag{6}$$

where $S_{i,i} = R_i S R_i^T$ is the ith edge operator and $S_{VV} = R_V S R_V^T$. The interaction between edges and between edges and vertices are neglected, hence the convergence rate deteriorates as $H \to 0$.

The Bramble-Pasciak-Schatz (BPS) preconditioner [BPS86] uses a *coarse grid operator* A_H to provide global coupling between subdomains. and an *interpolation map* $R_H^T : V \mapsto B$ is used to approximate the solution on B. The map $R_H : B \mapsto V$ is the *weighted restriction map* from B to V. The BPS preconditioner can be obtained by replacing the term $R_V^T S_{VV}^{-1} R_V$ in (6) with $R_H^T A_H^{-1} R_H$:

$$\hat{S}_{BPS}^{-1} = \sum_{i=1}^{n_E} R_i^T S_{i,i}^{-1} R_i + R_H^T A_H^{-1} R_H. \tag{7}$$

The condition number of $\hat{S}_{BPS}^{-1} S$ satisfies $\kappa(\hat{S}_{BPS}^{-1} S) = O\left(\log^2 (H/h)\right)$.

In a sense, the Chan-Mathew and BPS methods provide only strategies for constructing a preconditioner rather than an actual preconditioner. That is, they do not specify an efficient scheme for approximating $S_{i,i}$.

4 The Preconditioner

In this section we describe the edge preconditioner which is derived by applying a strip decomposition and approximating the Schur complement using a discretization on a special grid called the *edge grid*. Since we use strip decomposition, henceforth we replace the subscript B by E, to indicate that the interface contains only edges.

4.1 Edge Decomposition of the Schur Complement

The framework for constructing the edge preconditioner is the edge decomposition of S, described next. For strip substructuring, A_{EE} is a block diagonal matrix with one block per edge:

$$A_{EE} = \sum_{i=1}^{n_E} R_i^T A_{EE}^{(i)} R_i, \tag{8}$$

where the submatrix $A_{EE}^{(i)}$ represents the coefficients of the unknowns associated with edge i in the equations imposed at the collocation points associated with edge i.

Define $A_{EI}^{(i,j)}$ to be the matrix block which contains the coefficients of the unknowns from subdomain j in the equations imposed at the collocation points associated with edge i. The matrix A_{EI} has the decomposition:

$$A_{EI} = \sum_{i=1}^{n_E} R_i^T A_{EI}^{(i,i)} R_{I_i} + \sum_{i=1}^{n_E} R_i^T A_{EI}^{(i,i+1)} R_{I_{i+1}}, \tag{9}$$

where R_i, R_{I_i} are as defined in Subsection 3.1. Similarly, define $A_{IE}^{(i,j)}$ to be the matrix block which contains the coefficients of the unknowns from subdomain i in the equations imposed at the collocation points associated with edge j. The matrix A_{IE} has the decomposition:

$$A_{IE} = \sum_{i=1}^{n_E} R_{I_i}^T A_{IE}^{(i,i)} R_i + \sum_{i=1}^{n_E} R_{I_{i+1}}^T A_{IE}^{(i+1,i)} R_i. \tag{10}$$

Define the matrices $A_{EI}^{(i)}, A_{IE}^{(i)}$, and $A_{II}^{(i,i+1)}$ by $A_{EI}^{(i)} = \left(A_{EI}^{(i,i)}, A_{EI}^{(i,i+1)}\right)$, $A_{IE}^{(i)} = \left(A_{IE}^{(i,i)}, A_{IE}^{(i+1,i)}\right)^T$, and $A_{II}^{(i,i+1)} = Diag\left(A_{II}^{(i)}, A_{II}^{(i+1)}\right)$. The matrix form of the ith edge subproblem is:

$$A_{EI}^{(i)} \mathbf{u}_I^{(i,i+1)} + A_{EE}^{(i)} \mathbf{u}_E^{(i)} = \mathbf{f}_E^{(i)}. \tag{11}$$

The following lemma gives the structure of S for strip decomposition:

Lemma 1 *The Schur complement matrix S associated with a strip decomposition in n_E edges of size N_E is an $n_E \times n_E$ block tridiagonal matrix,*

$$S = \sum_{i=1}^{n_E} R_i^T S_{i,i} R_i + \sum_{i=1}^{n_E-1} R_i^T S_{i,i+1} R_{i+1} + \sum_{i=2}^{n_E} R_i^T S_{i,i-1} R_{i-1} \tag{12}$$

where $S_{i,i}, S_{i,i+1}, S_{i,i-1}$ are $N_E \times N_E$ matrices given by $S_{i,i} = A_{EE}^{(i)} - A_{EI}^{(i)} A_{II}^{(i,i+1)^{-1}} A_{IE}^{(i)}$, $S_{i,i+1} = -A_{EI}^{(i,i+1)} A_{II}^{(i+1)^{-1}} A_{IE}^{(i+1,i+1)}$, and $S_{i,i-1} = -A_{EI}^{(i,i)} A_{II}^{(i)^{-1}} A_{IE}^{(i,i-1)}$.

4.2 The Edge Preconditioner

We obtain an approximation for S by replacing (11) with the problem

$$\hat{A}_{EI}^{(i)} \mathbf{u}_I^{(i,i+1)} + \hat{A}_{EE}^{(i)} \mathbf{u}_E^{(i)} = \mathbf{f}_E^{(i)}. \tag{13}$$

We define an *edge neighborhood* $\mathcal{E}_e = [0,1] \times [y_e - H, y_e + H]$, for each edge $1 \leq e \leq n_E$, where H is an integer multiple of h and $y_e = eH + (e-1)h$. The edge grid associated with the eth edge is $G_e = X \times \{y_e - H, y_e, y_e + H\}$. Figure 1 shows the two edge grids for a three-strip decomposition. For each edge e, we approximate the restriction of u to $\mathcal{E}_e$ by an element in the function space of piecewise bicubic polynomials on $\mathcal{E}_e$, where the Hermite bicubic basis functions have support $[-h, h] \times [-H, H]$.

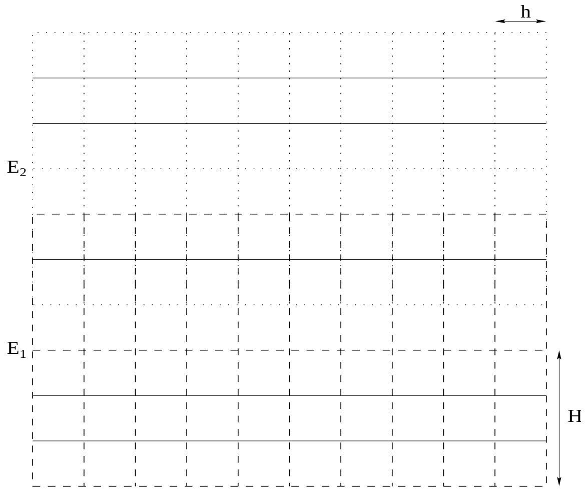


Figure 1: Edge grids for edge E_1 (dashed line) and edge E_2 (dotted line).

For each edge node, we collocate at the same points as for G^h, but in the new function space, and obtain (13). We approximate S in (4) by

$$\tilde{S} = \hat{A}_{EE} - \hat{A}_{EI} A_{II}^{-1} A_{IE}. \tag{14}$$

The structure of $\tilde{S}$ is given by Lemma 1. The effect of using the edge grid is to reduce the weight of the coefficients in the matrix $\hat{A}_{EI}$ as compared to the coefficients in $\hat{A}_{EE}$. For example, for the Laplacian operator it is easy to show that $\|\hat{A}_{EI}^{(i)}\|_\infty \approx \frac{h}{H} \|A_{EI}^{(i)}\|_\infty$ and $\|\hat{A}_{EE}^{(i)}\|_\infty = \|A_{EE}^{(i)}\|_\infty + O(1 - \frac{h}{H})$, where $1 \leq i \leq n_E$. The edge preconditioner $\hat{S}$ is derived from $\tilde{S}$ by making the approximations $\tilde{S}_{i,i} \approx \hat{A}_{EE}^{(i)}$, $\tilde{S}_{i,i+1} \approx 0$, $\tilde{S}_{i,i-1} \approx 0$. Thus, the edge preconditioner is:

$$\hat{S}^{-1} = \sum_{i=1}^{n_E} R_i^T \left(\hat{A}_{EE}^{(i)} \right)^{-1} R_i. \tag{15}$$

4.3 The System Preconditioner

The approximation of A^{-1} is obtained from (5) and (15)

$$M^{-1} = \begin{pmatrix} I & -A_{II}^{-1} A_{IE} \\ 0 & I \end{pmatrix} \begin{pmatrix} A_{II}^{-1} & 0 \\ 0 & \hat{S}^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -\hat{A}_{EI} A_{II}^{-1} & I \end{pmatrix}.$$

We determine the LU factorization of $A_{II}^{(i)}$, where $1 \le i \le n_I$, and of $A_{EE}^{(i)}$, where $1 \le i \le n_E$. The LU factorization is computed with scaled partial pivoting, based on the `BAND GE` module in the `ELLPACK` package [RB85], which we have adapted for blocking. The factored blocks are stored and used to perform triangular solves.

The application of the preconditioner, i.e., the computation of $\mathbf{v} = M^{-1}\mathbf{u}$ for a given $\mathbf{u} = (\mathbf{u}_I, \mathbf{u}_E)^T$, consists of a series of matrix-vector multiplications, triangular solves, and vector subtractions. A simpler version of the algorithm would start from step 4 with $\mathbf{y}_E = \mathbf{u}_E$.

Preconditioning Algorithm

1	$\mathbf{x}_I = A_{II}^{-1}\mathbf{u}_I;$	$\triangleright$ Solve harmonic problem
2	$\mathbf{x}_E = \hat{A}_{EI}\mathbf{x}_I;$	$\triangleright$ Correction for edge rhs
3	$\mathbf{y}_E = \mathbf{u}_E - \mathbf{x}_E;$	$\triangleright$ Update edge rhs
4	$\mathbf{v}_E = \hat{A}_{EE}^{-1}\mathbf{y}_E;$	$\triangleright$ Solve for edges
5	$\mathbf{y}_I = A_{IE}\mathbf{v}_E;$	$\triangleright$ Harmonic extension of $\mathbf{v}_E$
6	$\mathbf{v}_I = A_{II}^{-1}(\mathbf{u}_I - \mathbf{y}_I);$	$\triangleright$ Solve subdomains

Note that every step of the algorithm is fully parallel due to the block diagonal structure of A_{II} and $\hat{A}_{EE}$. Moreover, when the number of subdomains is $n_I = \Theta(\sqrt{N})$, the time complexity of the algorithm is $O(N)$.

5 Numerical Experiments

We evaluate the effectiveness of the preconditioner and compare it to block-Jacobi preconditioning. The performance metric is the number of iterations and the execution time of GMRES(k).

We consider the PDE ([RB85]: Problem 1, Appendix A), $e^{xy}u_{xx} + e^{-xy}u_{yy} + ye^{xy}u_x - xe^{-xy}u_y - \frac{1}{1+x+y}u = g$, defined on the unit square where the right-hand side g is determined by the solution $u = \frac{3}{4}e^{xy}\sin(\pi x)\sin(\pi y)$. The boundary conditions are: $u = 0$ on $x = 1$; $u_n = \frac{3\pi}{4}\sin(\pi y)$, on $x = 0$; $u_n = \frac{3\pi}{4}\sin(\pi x)$ on $y = 0$, and $u_n = \frac{3\pi}{4}e^x\sin(\pi x)$ on $y = 1$. We choose n_0 such that all subdomains have the same size. Let K be number of horizontal grid lines in a subdomain. For n_I subdomains there are $n_E = n_I - 1$ edges. The relation between K, n_I, n_E is $n_0 - n_E = Kn_I$. The total number of horizontal grid lines for the edge-preconditioned method is $n_0 = (K+1)n_I - 1$. The grid for block Jacobi is parameterized by K as $n_0 = Kn_I$. We fix a value for n_I then scale the number of grid lines by changing K.

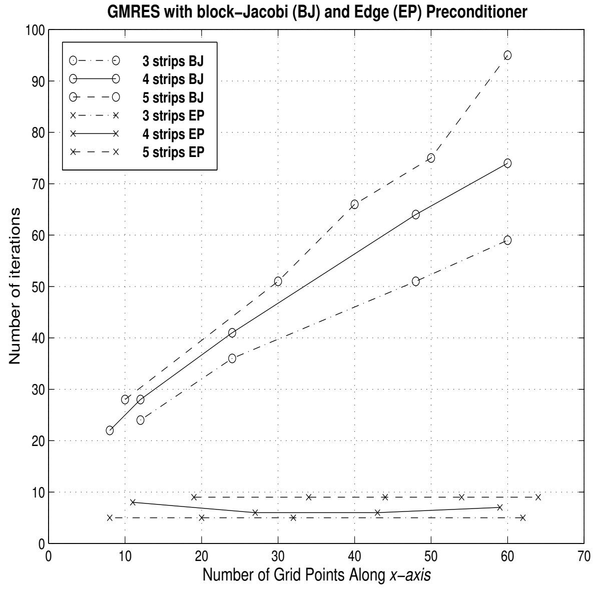
We employ GMRES(50), right-preconditioning, and the stopping rule is relative residual reduction of $\epsilon = 10^{-3}$. The results are given for $n_I = 3, 4, 5$ subdomains, with grid sizes varying between 8×8 and 64×64. The largest problem solved has 16384 unknowns.

The number of iterations and execution times for the block Jacobi and the edge preconditioner are shown in Figure 2. Unpreconditioned GMRES fails to converge. Our preconditioner is clearly superior to block Jacobi in terms of both numbers of iterations and execution time. We remark that the number of iterations for the edge preconditioner is low and independent of the grid diameter. For moderately fine grids and five subdomains, our method requires about ten times fewer iterations and its serial execution time is about ten times faster.

Thus, for a fixed number of subdomains, the number of iterations is virtually independent of the problem size. Furthermore, the number of iterations is small enough that the GMRES iteration does not have to be restarted, i.e., full GMRES is practical for the problems tested. Finally, for a fixed problem size, we see that the number of iterations grows extremely slowly with respect to the number of subdomains.

6 Concluding Remarks

We have introduced an interface preconditioner that is both accurate and efficient: accurate in that it is a good approximation to the original Schur complement, and efficient in that it is block diagonal, with one block corresponding to each interface subproblem. The preconditioner for A requires linear space and may provide an $O(N)$-time solution. Furthermore, all subdomains and edge subproblems can be dealt with in parallel.



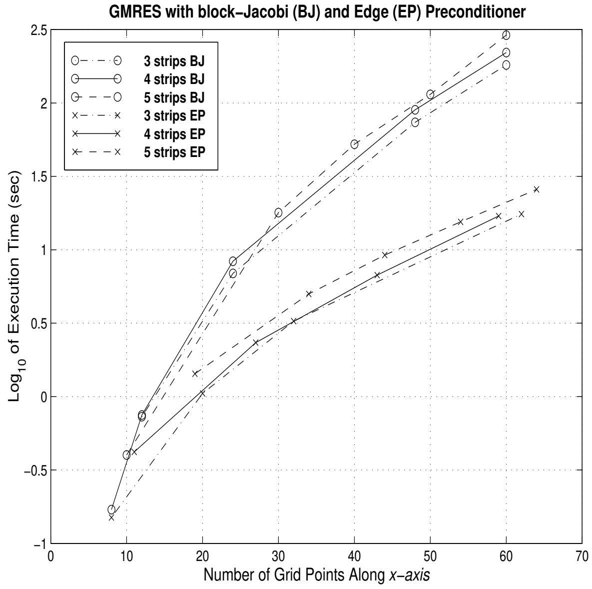


Figure 2: Iteration count and solver time for block-Jacobi and Edge Preconditioner

References

[BD93] B. Bialecki and M. Dryja. Multigrid and multilevel methods for orthogonal spline collocation problems. In *Copper Mountain Conference on Iterative Methods*, Copper Mt., CO., 1993.

[BF97] B. Bialecki and G. Fairweather. A piecewise Hermite bicubic orthogonal spline collocation method for the biharmonic dirichlet problem. In *Third IMACS Intl Symposium on Iterative Methods in Scientific Computation*, Jackson Hole, Wyoming, 1997.

[Bia97] B. Bialecki. A preconditionerd conjugate gradient method for orthogonal spline collocation problems. In *Third IMACS Intl Symposium on Iterative Methods in Scientific Computation*, Jackson Hole, Wyoming, 1997.

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

[BPX90] J. H. Bramble, J.E. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Mathematics of Computation*, 55:1–22, 1990.

[CM94] T. F. Chan and T.P. Mathew. Domain decomposition algorithms. *Acta Numerica*, pages 61–143, 1994.

[DR84] W.R. Dyksen and J.R. Rice. A new ordering scheme for the hermite bicubic collocation equations. In *In Elliptic Problem Solvers III*, pages 467–480. Academic Press, 1984.

[Hac92] W. Hackbusch. *Elliptic Diffferential Equations: Theory and Numerical Treatment*, volume 18 of *Springer Series in Computational Mathematics*. Springer-Verlag, 1992.

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

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

[SS86] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7:856–869, 1986.