Commun. Comput. Phys.
doi: 10.4208/cicp.080711.160212a

Vol. **13**, No. 2, pp. 411-427
February 2013

Additive Schwarz Preconditioners with Minimal Overlap for Triangular Spectral Elements

Yuen-Yick Kwan*

Center for Computational Science, Tulane University, New Orleans, LA 70118, USA.

Received 8 July 2011; Accepted (in revised version) 16 February 2012

Communicated by Jie Shen

Available online 28 June 2012

Abstract. The additive Schwarz preconditioner with minimal overlap is extended to triangular spectral elements (TSEM). The method is a generalization of the corresponding method in tensorial quadrilateral spectral elements (QSEM). The proposed preconditioners are based on partitioning the domain into overlapping subdomains, solving local problems on these subdomains and solving an additional coarse problem associated with the subdomain mesh. The results of numerical experiments show that the proposed preconditioner are robust with respect to the number of elements and are more efficient than the preconditioners with generous overlaps.

AMS subject classifications: 65F08, 65N35

Key words: Spectral elements, additive Schwarz.

1 Introduction

Spectral element method, which combines the flexibility of the low-order finite element methods and the high accuracy of the spectral methods, are popular in computational science and engineering. The efficiency of the spectral element method depends on the solution method employed to solve the resultant linear system. Since direct methods are infeasible when the number of elements get large, preconditioned iterative methods are usually used. Preconditioners can be based on an overlapping or non-overlapping domain decomposition [24,28]. The former include the multiplicative Schwarz [18], additive Schwarz [9] and the restricted Schwarz preconditioners [6]. The multiplicative Schwarz preconditioner generally has better convergence properties than the additive counterpart, but there is no straightforward way to parallelize it. The non-overlapping

*Corresponding author. *Email address:* tkwan@tulane.edu (Y.-Y. Kwan) preconditioners include Neumann-Neumann [10], Finite Element Tearing and Interconnect (FETI) [13], Balancing Domain Decomposition methods [20], Dual-Primal Finite Element Tearing and Interconnect (FETI-DP) [12], Balancing Domain Decomposition methods by Constraints (BDDC) [8] and optimized Schwarz [14, 22]. The preconditioned system from the FETI-DP and BDDC preconditioners have essentially the same spectrum [5,17,21]. Though most of these preconditioners are originally proposed for quadrilateral elements (QSEM), they can also be applied to triangular elements (TSEM). One exception is the additive Schwarz preconditioner with minimal overlap [7,19], as its construction relies heavily on the tensor-product nature of the collocation points.

In this paper, we construct additive Schwarz preconditioners with minimal overlap for TSEM. The construction is made possible by considering the Schur complement instead of the full matrix. We also compare the performance of the different preconditioners, including the Schwarz preconditioners with generous and minimal overlaps and the BDDC preconditioner. Four sets of collocation points are used in the numerical experiments, including the uniformly-distributed points, the Fekete points [2, 27], the Lobatto points [1] and the Lebesgue points [25].

The rest of the paper is organized as follows. Section 2 gives a brief review of spectral element method. Section 3 gives a brief review of the Schur complement and proves a theorem related to it. Section 4 introduces the different preconditioners and compares their computational complexities. Numerical experiments are presented in Section 5. Finally, some concluding remarks are given in Section 6.

2 Spectral element method

Consider the screened Poisson equation

$$\alpha u - \nabla^2 u = f \tag{2.1}$$

in $\Omega \subset \mathcal{R}^2$ with boundary conditions

$$u=0 \quad \text{on } \Gamma_{\mathrm{D}}, \qquad \frac{\partial u}{\partial n} = g_{\mathrm{N}} \quad \text{on } \Gamma_{\mathrm{N}}, \tag{2.2}$$

where α is a non-negative constant and $\Gamma_{\mathrm{D}} \cup \Gamma_{\mathrm{N}} = \partial\Omega$. Let $\mathcal{T} = \{T_i : i = 1, \cdots, N\}$ be a triangulation of the domain Ω, such that

$$\bigcup_{i=1}^{N} \overline{T_i} = \overline{\Omega}, \qquad T_i \cap T_j = \emptyset, \quad \text{for } i \neq j.$$

Then the spectral element space for the solution is

$$W_p(\mathcal{T}) := \mathcal{P}_p(\mathcal{T}) \cap W, \tag{2.3}$$

where

$$\mathcal{P}_p(\mathcal{T}) := \left\{ u \in \mathcal{C}^0(\Omega) : u|_{T_i} \in \mathcal{P}_p(T_i),\ \forall T_i \in \mathcal{T} \right\}$$

is the space of continuous functions whose restriction in each element T_i is a polynomial of degree at most p and

$$W = \{ u \in H^1(\Omega) : u = 0 \text{ on } \Gamma_{\mathrm{D}} \}$$

is the solution space to (2.1)-(2.2). The spectral element approximation to (2.1)-(2.2) is: Find $u \in W_p(\mathcal{T})$ such that

$$\int_\Omega (\alpha uv + \nabla u \cdot \nabla v) d\mathbf{x} = \int_\Omega fv d\mathbf{x} + \int_{\Gamma_{\mathrm{N}}} g_{\mathrm{N}} v ds \tag{2.4}$$

for all $v \in W_p(\mathcal{T})$, where the integrals on the right-hand side are approximated by replacing the functions f and g_{N} with interpolants. Once basis functions are chosen, (2.4) can be written as a linear system:

$$A\mathbf{u} = \mathbf{f}. \tag{2.5}$$

In practice, the integrals can be evaluated by expressing the basis functions as linear combinations of Dubiner polynomials [11]. More details on this can be found in [15].

Throughout this paper, Lagrange polynomials based on the collocation points are used as the basis functions.

3 Schur complement

Dividing the unknowns into those in the interior of elements ($\mathbf{u}_I$) and on the interface ($\mathbf{u}_\Gamma$), (2.5) can be written as:

$$\begin{bmatrix} A_{II} & A_{\Gamma I}^T \\ A_{\Gamma I} & A_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{u}_I \\ \mathbf{u}_\Gamma \end{bmatrix} = \begin{bmatrix} \mathbf{f}_I \\ \mathbf{f}_\Gamma \end{bmatrix}. \tag{3.1}$$

Using block Cholesky factorization, we have:

$$\begin{bmatrix} A_{II} & A_{\Gamma I}^T \\ A_{\Gamma I} & A_{\Gamma\Gamma} \end{bmatrix} = \begin{bmatrix} I & 0 \\ A_{\Gamma I} A_{II}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{II} & 0 \\ 0 & S_\Gamma \end{bmatrix} \begin{bmatrix} I & A_{II}^{-1} A_{\Gamma I}^T \\ 0 & I \end{bmatrix},$$

where $S_\Gamma = A_{\Gamma\Gamma} - A_{\Gamma I} A_{II}^{-1} A_{\Gamma I}^T$ is the Schur complement. Note that the Schur complement can be assembled from the element Schur complements:

$$S_\Gamma = \sum_{i=1}^N (R_\Gamma^{(i)})^T S_\Gamma^{(i)} R_\Gamma^{(i)}, \qquad S_\Gamma^{(i)} = A_{\Gamma\Gamma}^{(i)} - A_{\Gamma I}^{(i)} (A_{II}^{(i)})^{-1} (A_{\Gamma I}^{(i)})^T, \tag{3.2}$$

where $R_\Gamma^{(i)}$ is the restriction operator to Γ_i. Inverting the three factors, we have:

$$\begin{aligned} \begin{bmatrix} A_{II} & A_{\Gamma I}^T \\ A_{\Gamma I} & A_{\Gamma\Gamma} \end{bmatrix}^{-1} &= \begin{bmatrix} I & -A_{II}^{-1} A_{\Gamma I}^T \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{II}^{-1} & 0 \\ 0 & S_\Gamma^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{\Gamma I} A_{II}^{-1} & I \end{bmatrix} \\ &= R_I^T A_{II}^{-1} R_I + \Phi_\Gamma S_\Gamma^{-1} \Phi_\Gamma^T, \end{aligned} \tag{3.3}$$

where

$$\Phi_\Gamma = R_\Gamma^T - R_I^T A_{II}^{-1} A_{\Gamma I}^T,$$

and R_I, R_Γ are restriction operators to the interior and interface, respectively. Since $A_{II} = \mathrm{diag}(A_{II}^{(1)},\cdots,A_{II}^{(N)})$ is block diagonal, it is cheap to apply A_{II}^{-1} on a vector. Eq. (3.1) is then reduced to:

$$S_\Gamma \mathbf{u}_\Gamma = \mathbf{g}_\Gamma, \tag{3.4}$$

where $\mathbf{g}_\Gamma = \mathbf{f}_\Gamma - A_{\Gamma I} A_{II}^{-1} \mathbf{u}_I$.

There are two advantages in using the Schur complement. Firstly, the condition number of the Schur complement is likely to be much smaller than that of the full matrix [4]. Secondly, applying the Schur complement to a vector only requires $\mathcal{O}(Np^2)$ operations, while applying the full matrix to a vector requires $\mathcal{O}(Np^4)$ operations. As we shall see later, it is possible to construct the Schur complement in $\mathcal{O}(Np^4)$ operations. Hence when applying an iterative solution method, using the Schur complement is likely to be more efficient. Another interesting property is that when Lagrange polynomials based on collocation points are used as basis functions, the Schur complement is independent of the location of points in the interior of elements. This property is the direct consequence of the following more general theorem:

Theorem 3.1. *Let*

$$W_{p,0}(\mathcal{T}) = \left\{ u \in W_p(\mathcal{T}) : u = 0 \ \text{ on } \bigcup_{T\in\mathcal{T}} \partial T \right\}$$

be the set of functions in $W_p(\mathcal{T})$ *that vanish on element boundaries. Let* $\{\phi_i\}\cup\{\psi_i\}$ *and* $\{\tilde{\phi}_i\}\cup\{\tilde{\psi}_i\}$ *be two bases of* $W_p(\mathcal{T})$*. If* $\{\phi_i\}$ *and* $\{\tilde{\phi}_i\}$ *are bases of* $W_{p,0}(\mathcal{T})$ *and*

$$\psi_i - \tilde{\psi}_i \in W_{p,0}(\mathcal{T})$$

for all i, then the corresponding Schur complements are the same.

Proof. The full matrix and Schur complement correspond to the first basis are:

$$M = \begin{bmatrix} A & B^T \\ B & C \end{bmatrix}, \qquad S = C - BA^{-1}B^T,$$

where the entries of the matrices A, B and C are:

$$a_{ij} = \mathcal{L}(\phi_j,\phi_i), \qquad b_{ij} = \mathcal{L}(\phi_j,\psi_i), \qquad c_{ij} = \mathcal{L}(\psi_j,\psi_i),$$

with $\mathcal{L}$ the bilinear form in the variational formulation. The full matrix $\tilde{M}$ and the Schur complement $\tilde{S}$ correspond to the second basis are defined similarly. Since $\{\phi_i\}$ is a basis of $W_{p,0}(\mathcal{T})$, we may write:

$$\tilde{\phi}_i = \sum_k p_{ik}\phi_k, \qquad \tilde{\psi}_i = \psi_i + \sum_k q_{ik}\phi_k.$$

Then the entries of $\tilde{M}$ are:

$$
\begin{aligned}
\tilde{a}_{ij} &= \mathcal{L}(\tilde{\phi}_j,\tilde{\phi}_i) = \sum_{k,\ell} p_{jk} p_{i\ell} a_{k\ell} = (PAP^T)_{ij},\\
\tilde{b}_{ij} &= \mathcal{L}(\tilde{\phi}_j,\tilde{\psi}_i) = \sum_k p_{jk} b_{ik} + \sum_{k,\ell} p_{jk} q_{i\ell} a_{k\ell} = (BP^T + QAP^T)_{ij},\\
\tilde{c}_{ij} &= \mathcal{L}(\tilde{\psi}_j,\tilde{\psi}_i) = c_{ij} + \sum_k q_{jk} b_{ik} + \sum_\ell q_{i\ell} b_{j\ell} + \sum_{k,\ell} q_{jk} q_{i\ell} a_{k\ell} = (C + BQ^T + QB^T + QAQ^T)_{ij},
\end{aligned}
$$

and the Schur complement $\tilde{S}$ is:

$$
\begin{aligned}
\tilde{S} &= \tilde{C} - \tilde{B}\tilde{A}^{-1}\tilde{B}^T\\
&= C + BQ^T + QB^T + QAQ^T - (BP^T + QAP^T)(PAP^T)^{-1}(BP^T + QAP^T)^T\\
&= C + BQ^T + QB^T + QAQ^T - (B + QA)A^{-1}(B + QA)^T\\
&= C - BA^{-1}B^T.
\end{aligned}
$$

Hence the Schur complements are the same. □

Another consequence of the above theorem is that when computing the Schur complement, one can use any basis for $W_{p,0}(\mathcal{T})$. For example, by choosing a basis such that A_{II} in Eq. (3.1) is sparse, one can compute the Schur complement in $\mathcal{O}(Np^4)$ operations instead of $\mathcal{O}(Np^5)$.

4 Preconditioners

In this section, we describe the additive Schwarz preconditions with generous and minimal overlaps, as well as the BDDC preconditioner that will be used in the numerical experiments. We also compare the computational complexities of these preconditioners.

4.1 Additive Schwarz preconditioner

Let $\Omega_1,\cdots,\Omega_K$ be overlapping subdomains of Ω such that $\bigcup_{i=1}^K \overline{\Omega_i} = \overline{\Omega}$. The additive Schwarz preconditioner is:

$$
M_{\text{AS}}^{-1} = R_0^T (R_0 A R_0^T)^{-1} R_0 + \sum_{i=1}^{K} (R_D^{(i)})^T (R^{(i)} A (R^{(i)})^T)^{-1} R_D^{(i)}, \tag{4.1}
$$

where R_0^T is the interpolation operator from the coarse grid to the fine grid with R_0 its transpose, $R^{(i)}$ is the restriction operator to the interior of Ω_i and $R_D^{(i)}$ is either the same as $R^{(i)}$ or a scaled version of it.

The overlapping subdomains can be based on a partitioning of vertices or elements. If the vertices are partitioned into sets $\mathcal{V}_1,\cdots,\mathcal{V}_K$, then the overlapping subdomains can

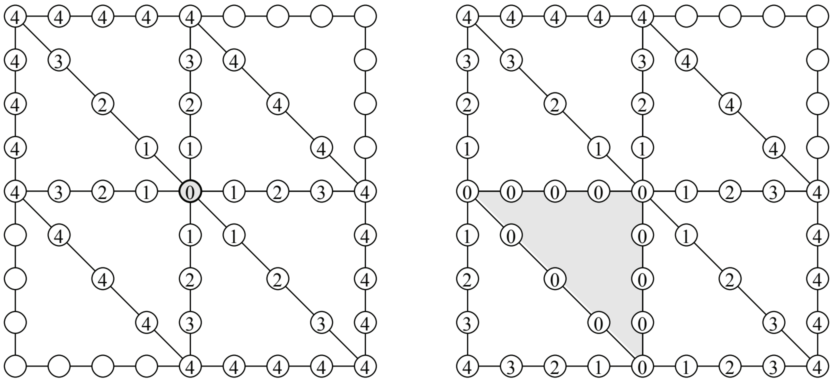

Figure 1: Distance function of a node from a vertex (left) and from an element (right) in the case $p=4$. The distance is infinity for nodes without a number.

be constructed such that $\mathcal{V}_i$ is in the interior of subdomain Ω_i. With generous overlap, Ω_i is the union of elements with at least one vertex in $\mathcal{V}_i$ [26]:

$$\Omega_i=\bigcup\{T\in\mathcal{T}:\mathcal{V}(T)\cap\mathcal{V}_i\neq\emptyset\}, \tag{4.2}$$

where $\mathcal{V}(T)$ is the set of vertices in element T. To construct subdomains with minimal overlap, we need to define the distance from a node to a vertex. The definition of the distance function is not clear for non-tensorial mesh. However, if preconditioner is applied to the Schur complement instead of the full matrix, then we only need to define the distance for nodes on the element interfaces, as shown in Fig. 1(left). With the distance function defined, with minimal overlap, Ω_i contains the nodes that are at most a little more than "half" an element away from $\mathcal{V}_i$:

$$\Omega_i=\left\{\mathbf{x}_k:\min_{V\in\mathcal{V}_i}\text{dist}(\mathbf{x}_k,V)\leq\left\lfloor\frac{p}{2}\right\rfloor+2\right\}. \tag{4.3}$$

If the elements are partitioned into sets $\mathcal{T}_1,\cdots,\mathcal{T}_K$, then the overlapping subdomains can be constructed such that $\mathcal{T}_i$ is in the interior of subdomain Ω_i. With generous overlap, Ω_i is just the union of the elements in $\mathcal{T}_i$:

$$\Omega_i=\bigcup\{T\in\mathcal{T}_i\}. \tag{4.4}$$

With minimal overlap, Ω_i contains the nodes that are at most two nodes away from $\mathcal{T}_i$:

$$\Omega_i=\left\{\mathbf{x}_k:\min_{T\in\mathcal{T}_i}\text{dist}(\mathbf{x}_k,T)\leq 2\right\}, \tag{4.5}$$

where the distance function from a node to an element is shown in Fig. 1(right).

4.2 BDDC preconditioner

In the BDDC preconditioner, the subdomain unknowns are divided into two types: the primal variables $\mathbf{u}_\Pi^{(i)}$, which are continuous across interface and the dual variables $\mathbf{u}_\Delta^{(i)}$, which are discontinuous across interface. Several restriction operators need to be defined. $R_\Pi^{(i)}$ and $R_\Delta^{(i)}$ are restriction operators to the subdomain primal and dual variables, respectively. R_Π is the direct sum of the $R_\Pi^{(i)}$'s and R_Δ is the direct sum of the $R_\Delta^{(i)}$'s. $R_{D,\Pi}$ and $R_{D,\Delta}$ are scaled versions of R_Π and R_Δ, defined as:

$$R_{D,\Pi}=R_\Pi C^{-1},\qquad R_{D,\Delta}=R_\Delta C^{-1};\qquad C=R_\Pi^T R_\Pi+R_\Delta^T R_\Delta.$$

$R_{D,\Pi}^{(i)}$ and $R_{D,\Delta}^{(i)}$ are the restrictions of $R_{D,\Pi}$ and $R_{D,\Delta}$ to subdomain i, respectively. R_D is the direct sum of R_Π and $R_{D,\Delta}$.

The action of the preconditioner on a vector, $M_{\text{BDDC}}^{-1}\mathbf{f}=\mathbf{u}$, is computed as follows. First form the subdomain linear systems:

$$\begin{bmatrix} A_{\Delta\Delta}^{(i)} & (A_{\Pi\Delta}^{(i)})^T \\ A_{\Pi\Delta}^{(i)} & A_{\Pi\Pi}^{(i)} \end{bmatrix}\begin{bmatrix}\mathbf{u}_\Delta^{(i)}\\ \mathbf{u}_\Pi^{(i)}\end{bmatrix}=\begin{bmatrix}\mathbf{f}_\Delta^{(i)}\\ \mathbf{f}_\Pi^{(i)}\end{bmatrix},$$

where $\mathbf{f}_\Delta^{(i)}=R_{D,\Delta}^{(i)}\mathbf{f}$ and $\mathbf{f}_\Pi^{(i)}=R_{D,\Pi}^{(i)}\mathbf{f}$. Assembling only the primal variables results in the linear system:

$$\begin{bmatrix} A_{\Delta\Delta} & A_{\Pi\Delta}^T \\ A_{\Pi\Delta} & A_{\Pi\Pi} \end{bmatrix}\begin{bmatrix}\mathbf{u}_\Delta\\ \mathbf{u}_\Pi\end{bmatrix}=\begin{bmatrix}\mathbf{f}_\Delta\\ \mathbf{f}_\Pi\end{bmatrix},$$

where

$$A_{\Pi\Pi}=\sum_{i=1}^N (R_\Pi^{(i)})^T A_{\Pi\Pi}^{(i)} R_\Pi^{(i)},\qquad \mathbf{f}_\Pi=\sum_{i=1}^N (R_\Pi^{(i)})^T \mathbf{f}_\Pi^{(i)}=R_\Pi\mathbf{f}.$$

The above system can be inverted as in Section 3 to give:

$$\begin{bmatrix} A_{\Delta\Delta} & A_{\Pi\Delta}^T \\ A_{\Pi\Delta} & A_{\Pi\Pi} \end{bmatrix}^{-1}=R_\Delta^T A_{\Delta\Delta}^{-1} R_\Delta+\Phi_\Pi S_\Pi^{-1}\Phi_\Pi^T,$$

where

$$\Phi_\Pi=R_\Pi^T-R_\Delta^T(A_{\Delta\Delta})^{-1}A_{\Pi\Delta}^T=R_\Pi^T-R_\Delta^T\sum_{i=1}^N (R_\Delta^{(i)})^T(A_{\Delta\Delta}^{(i)})^{-1}(A_{\Pi\Delta}^{(i)})^T R_\Pi^{(i)},$$

and S_Π can be assembled from the subdomain matrices similar to in (3.2). Finally, $\mathbf{u}$ is computed as $R_\Pi^T u_\Pi+R_{D,\Delta}^T\mathbf{u}_\Delta$. In summary, the BDDC preconditioner can be written as:

$$M_{\text{BDDC}}^{-1}=R_D^T\big(R_\Delta^T A_{\Delta\Delta}^{-1}R_\Delta+\Phi_\Pi S_\Pi^{-1}\Phi_\Pi^T\big)R_D. \tag{4.6}$$

4.3 Complexity estimates

We examine here the computational costs of solving the Schur system (3.4) using different preconditioners. Table 1 summarizes the different preconditioners. For simplicity, we only consider the case with maximum number of subdomains. In other words, the number of subdomains of the Schwarz(E1), Schwarz(E2) and BDDC preconditioners equals the number of elements N and the number of subdomains of the Schwarz(V1) and Schwarz(V2) preconditioners equals the number of vertices $|\mathcal{V}|$.

Table 1: Summary of preconditioners.

Name	Preconditioner
Schwarz(V1)	Additive Schwarz based on vertices (generous overlap)
Schwarz(V2)	Additive Schwarz based on vertices (minimal overlap)
Schwarz(E1)	Additive Schwarz based on elements (generous overlap)
Schwarz(E2)	Additive Schwarz based on elements (minimal overlap)
BDDC	BDDC preconditioner

In each iteration, there are a coarse solver, some matrix-vector products and the local solvers. In the additive Schwarz preconditioners, the coarse solvers corresponds to the operator $(R_0AR_0^T)^{-1}$ in (3.2), while in the BDDC preconditioner, the coarse solver corresponds to the operator $A_{\Delta\Delta}^{-1}$ in (4.6). There are several ways to solve the coarse problem, including banded Cholesky factorization, multigrid iteration and approximate solver [29], to name a few. Finding the best estimate of the cost of the coarse solver is thus very complicated and beyond the scope of the paper. Next, as mentioned in Section 3, the cost of applying the Schur complement to a vector is $\mathcal{O}(Np^2)$. Finally, we assume that the local matrices have been factored before starting the iterations, so that the cost of the local solvers is $\mathcal{O}(\sum_{i=1}^K d_i^2)$, where K is the number of subdomains and d_i is the number of unknowns in subdomain i. In all preconditioners considered, the number of subdomains is $\mathcal{O}(N)$ and the number of unknowns per subdomain is $\mathcal{O}(p)$. Hence the total cost per iteration is $\mathcal{O}(Np^2)$ plus the cost of the coarse solver.

It may seem that the preconditioners have the same cost per iteration. Note, however, that the actual number of subdomains and number of unknowns per subdomain are different among the preconditioners. The number of subdomains of the Schwarz(V1) and Schwarz(V2) preconditioners equals the number of vertices $|\mathcal{V}|$, which is related to N as follows. The sum of the interior angles of the elements is $N\pi$. The sum of the angles in the interior of the domain is $2|\mathcal{V}_I|\pi$, where $\mathcal{V}_I$ is the set of vertices in the interior of the domain. So we have the relation:

$$2|\mathcal{V}_I|\pi \leq N\pi \leq 2|\mathcal{V}|\pi.$$

In most cases the number of interior vertices is close to the total number of vertices, i.e., $|\mathcal{V}_I| \approx |\mathcal{V}|$. Combining this with the above relation gives $|\mathcal{V}| \approx 0.5N$. Next, the number of unknowns per subdomain d_i is estimated as follows. In average each interior vertex has six edges linking to it, each with $p-1$ unknowns. Hence we have $d_i \approx 6p$ for the

Table 2: Rough estimates of the cost of local solvers.

Method	No. of subdomains (K)	Unknowns per subdomain (d_i)	Kd_i^2
Schwarz(V1)	$0.5N$	$6p$	$18Np^2$
Schwarz(V2)	$0.5N$	$3p$	$4.5Np^2$
Schwarz(E1)	N	$15p$	$225Np^2$
Schwarz(E2)	N	$3p$	$9Np^2$
BDDC	N	$3p$	$9Np^2$

Schwarz(V1) preconditioner. For the Schwarz(V2) preconditioner, each edge only has around $p/2$ unknowns belonging to the subdomain. Hence we have $d_i \approx 3p$. For the Schwarz(E1) preconditioner, the number of interior edges per subdomain can be counted as follows. Each subdomain has three vertices, each of which has in average six edges linking to it. This way the edges of the center element are each counted twice. Hence the number of interior edges per subdomain is approximately $6(3)-3=15$. Thus we have $d_i \approx 15p$. Finally, for the Schwarz(E2) preconditioner, except for the edges of the center elements, the other interior edges each only has one unknown belonging to the subdomain. Hence we have $d_i \approx 3p$. Table 2 summarizes the estimates of the cost of the local solvers in the different preconditioners.

When using preconditioned conjugate gradient (PCG), the number of iterations is proportion to the square root of the condition number. Hence to estimate the total cost of solving the linear system, we also need to estimate the condition number of the preconditioned system. Unfortunately, the existing theory is very limited. In the case of QSEM, we have the following results [16,23]:

$$\begin{aligned} &\text{Schwarz(E1)}: && \kappa_2(M^{-1}A) \leq C, \\ &\text{Schwarz(E2)}: && \kappa_2(M^{-1}A) \leq C(1+p^2), \\ &\text{BDDC}: && \kappa_2(M^{-1}A) \leq C(1+\log p)^2, \end{aligned}$$

with $C>0$ independent of the polynomial degree p and the number of elements N. In the case of TSEM, the only known result is the following [26]:

$$\text{Schwarz(V1)}: \quad \kappa_2(M^{-1}A) \leq C.$$

Note that in the above, the results concerning overlapping Schwarz preconditioners are for the full matrix. To the author's best knowledge, there is no related theoretical work when the Schur complement is used.

5 Numerical results

Throughout the section, Eq. (2.1) in $\Omega=(-1,1)^2$ is solved with exact solution

$$u(x,y)=\sin(\pi x)\sin(\pi y).$$

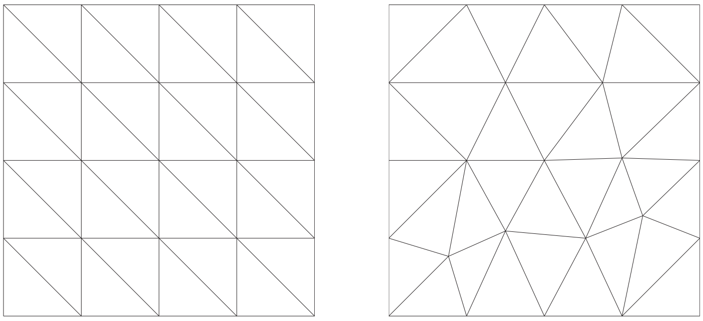

Figure 2: Structured (left) and unstructured (right) triangulations of the computational domain.

Unless otherwise mentioned, $\alpha=0$ and Dirichlet boundary conditions are used. The domain is divided into 32 structured triangles, as shown in Fig. 2 and refinements are constructed by dividing each triangle into four smaller ones. Also shown in the figure is an unstructured triangulation that is used in some comparisons.

5.1 Collocation points

The collocation points used in the numerical experiments include the uniform points, the Fekete points [2, 27], the Lobatto points [1] and the Lebesgue points [25]. The uniform points are constructed by placing the points uniformly in the reference triangle $\hat{T}$:

$$\Big(\frac{i}{p},\frac{j}{p}\Big), \qquad i=0,\cdots,p; \quad j=0,\cdots,p-i. \tag{5.1}$$

The Fekete points are constructed by maximizing the determinant of the Vandermonde matrix:

$$V=\begin{bmatrix} b_1(x_1,y_1) & \cdots & b_1(x_m,y_m) \\ \vdots & \ddots & \vdots \\ b_m(x_1,y_1) & \cdots & b_m(x_m,y_m) \end{bmatrix},$$

where $m=(p+1)(p+2)/2$ is the number of collocation points, $\{b_i\}$ is a basis of $\mathcal{P}_p(\hat{T})$ and the (x_i,y_i)'s are the collocation points. Fekete points on an interval and on the square are just Gauss-Lobatto points [3]. Moreover, in all known cases (polynomial degree up to 18) the Fekete points lying on the edges of an triangle are one-dimensional Gauss-Lobatto points.

The Lobatto points are constructed from one-dimensional Gauss-Lobatto points. If $\{\xi\}$ is the set of uniform points on [0,1], then the uniform points in (5.1) are actually the intersections of the lines $x=\xi_i$, $y=\xi_j$ and $x+y=\xi_{i+j}$. If $\{\xi_i\}$ is the set of Gauss-Lobatto points, then the three lines do not intersect and thus form a triangle. The Lobatto points are the centroids of the triangles formed this way. They are given by the formula:

$$\Big(\frac{2\xi+\xi_{i+j}-\xi_j}{3},\frac{2\xi_j+\xi_{i+j}-\xi_i}{3}\Big), \qquad i=0,\cdots,p; \quad j=0,\cdots,p-i. \tag{5.2}$$

The Lebesgue points are obtained by minimizing the Lebesgue constant:

$$\Lambda = \max_{(x,y)\in\hat{T}} \sum_{i=1}^{m} |\ell_i(x,y)|,$$

where the ℓ_i's are the Lagrange polynomials based on the collocation points.

The different sets of collocation points in the case $p=6$ are shown in Fig. 3.

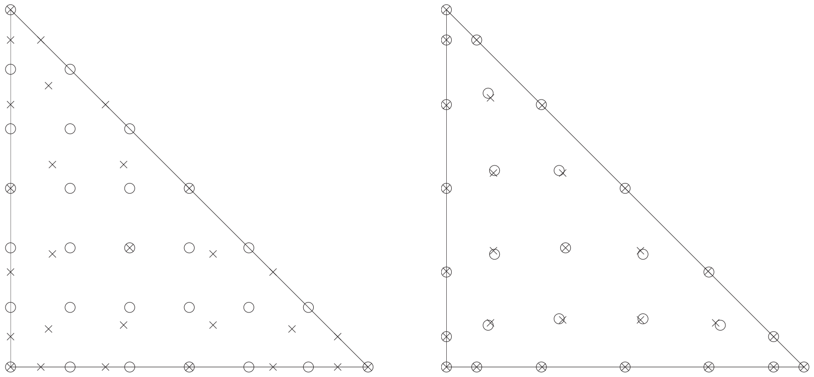

Figure 3: Different collocation points for $p=6$. Left: Uniform ($\circ$) and Fekete ($\times$) points. Right: Lobatto ($\circ$) and Lebesgue ($\times$) points.

Table 3 shows the condition numbers of the full matrix and the Schur complement in the case $N=32$. Lagrange polynomials based on the collocation points are used as the basis functions. Note that the condition number for the full matrix is much larger than that of the Schur complement. This suggests that a solution method, direct or iterative, should be applied to the Schur complement instead of the full matrix, because of the reduced problem size and condition number. Note also that the Fekete and Lobatto points, which coincide on the interface, give the same condition numbers for the Schur complement. This is because, as shown in Section 3, the condition number of the Schur complement depends only on the collocation points on the interface.

Table 3: Condition number estimates for the full matrix and the Schur complement.

Points	Full					Schur				
	$p=4$	$p=6$	$p=8$	$p=10$	$p=12$	$p=4$	$p=6$	$p=8$	$p=10$	$p=12$
Uniform	3.2E2	2.7E3	2.6E4	3.1E5	4.0E6	1.6E2	8.3E2	5.4E3	4.5E4	4.5E5
Fekete	2.3E2	8.8E2	3.0E3	5.5E3	1.1E4	8.2E1	1.4E2	2.0E2	2.6E2	3.2E2
Lobatto	2.3E2	9.8E2	3.8E3	1.6E4	8.0E4	8.2E1	1.4E2	2.0E2	2.6E2	3.2E2
Lebesgue	2.5E2	9.8E2	3.5E3	5.5E3	1.1E4	8.2E1	1.4E2	1.9E2	2.6E2	3.2E2

5.2 Comparison of preconditioners

Next we compare the performance of the different preconditioners for the Schur complement. For the Schwarz(V1) and Schwarz(V2) preconditioners we take $R_D^{(i)} = R^{(i)}$ in (4.1), while for the Schwarz(E1) and Schwarz(E2) preconditioners it is found that better results are obtained by taking the scaled version suggested in [19]:

$$R_D^{(i)} = R^{(i)} C^{-\frac{1}{2}}, \qquad C = \sum_{i=1}^{N} (R^{(i)})^T R^{(i)}.$$

Two BDDC preconditioners are considered. The first one, BDDC(V), has only the vertex variables as primal variables. The second one, BDDC(VE), has both the vertex variables and edge averages as primal variables [17]. Preconditioners with coarse problem of the same size are compared with each other. The level-1 conditioners are those with a smaller coarse problem, which includes the Schwarz preconditioners with coarse problem $p=1$ and the BDDC(V) preconditioner. On the other hand, the level-2 preconditioners includes Schwarz preconditioners with coarse problem $p=2$ and the BDDC(VE) preconditioner.

Eq. (2.1) with $\alpha = 0$ is solved with Dirichlet boundary conditions and a structured triangulation. The Schur system (3.4) is solved by the PCG method with zero initial guess. Tables 4 and 5 show the iteration counts to reduce the relative residual to 10^{-7} for fixed polynomial degree $p = 12$. The iteration counts for the Lobatto and Lebesgue points are not shown as they are very similar to those for the Fekete points. The following observations are made from the two tables: (1) The iteration count is independent of the number of triangles except for the BDDC(V) preconditioner; (2) Uniform points take more iterations than the other sets of points, especially for the Schwarz(E2) preconditioner; (3) All level-2 preconditioners take similar number of iterations when the uniform points are not used.

Tables 6 and 7 show the iteration counts for fixed number of triangles $N=2048$. From the tables, the Schwarz(V1) and Schwarz(E1) preconditioners are robust with respect to

Table 4: Iteration counts for different number of triangles N and fixed polynomial degree $p = 12$: level-1 preconditioners.

Method	Points	$N=32$	$N=128$	$N=512$	$N=2048$
Schwarz(V1)	Uniform	16	18	18	17
	Fekete	14	15	15	15
Schwarz(V2)	Uniform	20	21	20	20
	Fekete	15	17	16	16
Schwarz(E1)	Uniform	14	15	15	15
	Fekete	13	14	14	14
Schwarz(E2)	Uniform	26	28	27	26
	Fekete	16	16	16	15
BDDC(V)	Uniform	22	32	34	35
	Fekete	18	28	30	31

Table 5: Iteration counts for different number of triangles N and fixed polynomial degree $p=12$: level 2 coarse problem.

Method	Points	$N=32$	$N=128$	$N=512$	$N=2048$
Schwarz(V1)	Uniform	16	16	16	15
	Fekete	14	14	13	13
Schwarz(V2)	Uniform	17	17	15	15
	Fekete	13	13	13	12
Schwarz(E1)	Uniform	14	14	14	13
	Fekete	12	13	12	12
Schwarz(E2)	Uniform	25	25	23	23
	Fekete	13	13	12	12
BDDC(VE)	Uniform	14	16	16	16
	Fekete	13	14	14	14

Table 6: Iteration counts for different polynomial degrees p and fixed number of triangles $N=2048$: level-1 preconditioners.

Method	Points	$p=4$	$p=6$	$p=8$	$p=10$	$p=12$
Schwarz(V1)	Uniform	15	15	16	16	17
	Fekete	15	15	15	15	15
Schwarz(V2)	Uniform	15	16	18	19	20
	Fekete	15	15	15	16	16
Schwarz(E1)	Uniform	13	13	13	14	15
	Fekete	12	13	13	14	14
Schwarz(E2)	Uniform	15	17	19	23	26
	Fekete	13	14	14	14	15
BDDC(V)	Uniform	20	24	28	32	35
	Fekete	20	23	26	29	31

Table 7: Iteration counts for different polynomial degrees p and fixed number of triangles $N=2048$: level-2 preconditioners.

Method	Points	$p=4$	$p=6$	$p=8$	$p=10$	$p=12$
Schwarz(V1)	Uniform	12	13	14	15	15
	Fekete	12	13	13	12	13
Schwarz(V2)	Uniform	12	12	14	15	15
	Fekete	12	13	13	13	12
Schwarz(E1)	Uniform	10	11	12	13	13
	Fekete	10	11	11	12	12
Schwarz(E2)	Uniform	12	14	16	20	23
	Fekete	9	11	12	12	12
BDDC(VE)	Uniform	8	11	13	15	16
	Fekete	8	10	12	13	14

Table 8: Iteration counts under different conditions with fixed polynomial degree $p=12$ and number of triangles $N=2048$: level-1 preconditioners.

Method	Points	Condition				
		Original	$\alpha=1$	Neumann	Random	Unstructured
Schwarz(V1)	Uniform	17	17	19	18	18
	Fekete	15	15	18	16	15
Schwarz(V2)	Uniform	20	20	22	22	22
	Fekete	16	16	19	17	17
Schwarz(E1)	Uniform	15	15	18	16	16
	Fekete	14	14	19	14	14
Schwarz(E2)	Uniform	26	26	27	28	29
	Fekete	15	15	19	17	16
BDDC(V)	Uniform	35	35	34	37	40
	Fekete	31	30	32	32	34

Table 9: Iteration counts under different conditions with fixed polynomial degree $p=12$ and number of triangles $N=2048$: level-2 preconditioners.

Method	Points	Condition				
		Original	$\alpha=1$	Neumann	Random	Unstructured
Schwarz(V1)	Uniform	15	15	19	18	18
	Fekete	13	13	18	15	15
Schwarz(V2)	Uniform	15	15	17	17	17
	Fekete	12	12	16	14	14
Schwarz(E1)	Uniform	13	13	17	15	15
	Fekete	12	12	16	13	13
Schwarz(E2)	Uniform	23	23	24	25	25
	Fekete	12	12	16	13	14
BDDC(VE)	Uniform	16	16	15	17	19
	Fekete	14	14	14	14	16

number of triangles and polynomial degree. The Schwarz(V2) preconditioner is robust when the uniform points are not used, but the iteration number depends weakly on the polynomial degree when the uniform points are used. The Schwarz(E2) preconditioner behaves similarly, except that the iteration number grows faster when uniform points are used. For the BDDC(V) and BDDC(VE) preconditioners, the iteration number increases no matter which set of points are used.

Tables 8 and 9 show the iteration counts under different conditions, including: (a) original setting as in the previous comparisons, (b) with $\alpha=1$ in (2.1), (c) using Neumann boundary conditions instead of Dirichlet ones, (d) solving with a random right-hand side and (e) using the unstructured triangulation. Only the results with number of triangles $N=2048$ and polynomial degree $p=12$ are shown. From the tables, all methods are robust to the conditions considered.

6 Conclusions

We have extended the additive Schwarz preconditioner with minimal overlap to TSEM. The proposed preconditioners are applied to the Schur complement instead of the full matrix, because the former has a smaller size and smaller condition number (Table 3). It is proved that under certain conditions different bases would lead to the same Schur complement (Theorem 3.1). In particular, when Lagrange polynomials based on collocation points are used as basis function, the Schur complement is independent of the location of points in the interior of elements.

Different preconditioners for the Schur complement are compared. Collocation points used include uniform points, Fekete points, Lobatto points and Lebesgue points. The uniform points are inferior to the other sets of points for two reasons. Firstly, the condition number of the resultant linear system grows much faster (Table 3). Secondly, for most of the preconditioners considered, more iterations are needed when the uniform points are used (Tables 4-7). The Schwarz(V1) and Schwarz(E1) preconditioners are robust with respect to number of triangles and polynomial degree. It is not surprising that the Schwarz(E1) preconditioner gives fewer iteration counts than the Schwarz(V1) preconditioner, as in the Schwarz(E1) preconditioner adjacent subdomains have a bigger overlap than in the Schwarz(V1) preconditioner. The Schwarz(V2) and Schwarz(E2) preconditioners are robust when the uniform points are not used, but the iteration count increases with the polynomial degree when the uniform points are used. For the BDDC(V) and BDDC(VE) preconditioners, the iteration count increases with the polynomial degree, no matter which set of points are used. The BDDC(V) preconditioner is the only one that the iteration count increases with number of triangles.

The Schwarz preconditioners with minimal overlap are promising in solving large linear systems resulting from TSEM. They require less work per iteration than the corresponding preconditioners with generous overlap, yet the iteration count is similar when the uniform points are not used. Unfortunately, it may not be easy to prove analytically the growth of the condition number for these preconditioners, as the growth seems to depend on the collocation points used.

Acknowledgments

We would like to thank Prof. Mark Taylor for his computed Fekete points and Michael Roth for his computed Lebesgue points. This work is partially supported by NSF grant EPS-0701491.

References

[1] M. G. Blyth and C. Pozrikids, A Lobatto interpolation grid over the triangle, IMA J. Appl. Math., 71(1) (2006), 153–169.

[2] L. Bos, Bounding the Lebesgue function for Lagrange interpolation in a simplex, J. Approx. Theory, 38 (1983), 43–59.

[3] L. Bos, M. A. Taylor and B. A. Wingate, Tensor product Gauss-Lobatto points are Fekete points for the cube, Math. Comput., 70 (2001), 1543–1547.

[4] S. C. Brenner, The condition number of the Schur complement in domain decomposition, Numer. Math., 83(2) (1999), 187–203.

[5] S. C. Brenner and L.-Y. Sung, BDDC and FETI-DP without matrices or vectors, Comput. Methods Appl. Mech. Eng., 196(8) (2007), 1429–1435.

[6] X.-C. Cai and M. Sarkis, A restricted additive Schwarz preconditioner for general sparse linear systems, SIAM J. Sci. Comput., 21(2) (1999), 792–797.

[7] M. A. Casarin, Quasi-optimal Schwarz methods for the conforming spectral element discretization, SIAM J. Numer. Anal., 34(6) (1997), 2482–2502.

[8] C. R. Dohrmann, A preconditioner for substructuring based on constrained energy minimization, SIAM J. Sci. Comput., 25(1) (2003), 246–258.

[9] M. Dryja and O. B. Widlund, An additive variant of the Schwarz alternating method in the case of many subregions, Technical Report 339, Department of Computer Science, Courant Institute, 1987.

[10] M. Dryja and O. B. Widlund, Towards a unified theory of domain decomposition algorithms for elliptic problems, in T. F. Chan, R. Glowinski, J. Périaux and O. B. Widlund, editors, Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, pages 3–21, Society for Industrial and Applied Mathematics, 1989.

[11] M. Dubiner, Spectral methods on triangles and other domains, J. Sci. Comput., 6(4) (1991), 345–390.

[12] C. Farhat, M. Lesoinne, P. Le Tallec, K. Pierson and D. Rixen, FETI-DP: a dual-primal unified FETI method-part I: a faster alternative to the two-level FETI method, Int. J. Numer. Meth. Eng., 50(7), 1523–1544.

[13] C. Farhat and F.-X. Roux, A method of finite element tearing and interconnecting and its parallel solution algorithm, Int. J. Numer. Meth. Eng., 32(6), 1205–1227.

[14] M. J. Gander, Optimized Schwarz methods, SIAM J. Numer. Anal., 44(2) (2006), 699–731.

[15] K. C. Kirby, Algorithm 839: FIAT, a new paradigm for computing finite element basis functions, ACM Trans. Math. Software, 30(4) (2004), 502–516.

[16] A. Klawonn, L. F. Pavarino and O. Rheinbach, Spectral element FETI-DP and BDDC preconditioners with multi-element subdomains, Comput. Methods Appl. Mech. Eng., 198 (2008), 511–523.

[17] J. Li and O. B. Widlund, FETI-DP, BDDC and block Cholesky methods, Int. J. Numer. Meth. Eng., 66 (2006), 250–271.

[18] P. L. Lions, On the Schwarz method I, in R. Glowinski, G. H. Golub, G. A. Meurant and J. Périaux, editors, First International Symposium on Domain Decomposition Methods for Partial Differential Equations, pages 1–42, Society for Industrial and Applied Mathematics, 1988.

[19] J. W. Lottes and P. F. Fischer, Hybrid multigrid/Schwarz algorithms for the spectral element method, J. Sci. Comput., 24(1) (2005), 45–78.

[20] J. Mandel, Balancing domain decomposition, Commun. Numer. Methods Eng., 9(3) (1993), 233–241.

[21] J. Mandel, C. R. Dohrmann and R. Tezaur, An algebraic theory for primal and dual substructuring methods by constraints, Appl. Numer. Math., 54(2) (2005), 167–193.

[22] F. Nataf, F. Rogier and E. de Sturler, Optimal interface conditions for domain decomposition

methods, Technical Report 301, CMAP, Ecole Polytechnique, 1994.

[23] L. F. Pavarino, E. Zampieri, R. Pasquetti and F. Rapetti, Overlapping Schwarz methods for Fekete and Gauss-Lobatto spectral elements, SIAM J. Sci. Comput., 29(3) (2007), 1073–1092.

[24] A. Quarteroni and A. Valli, Domain Decomposition Methods for Partial Differential Equations, New York: Oxford University Press, 1999.

[25] M. J. Roth, Nodal Configurations and Voronoi Tessellations for Triangular Spectral Elements, Ph.D. thesis, University of Victoria, Victoria, BC, Canada, 2005.

[26] J. Schöberl, J. M. Melenk, C. Pechstein and S. Zaglmayr, Additive Schwarz preconditioning for p-version triangular and tetrahedral finite elements, IMA J. Appl. Numer. Anal., 28(1) (2008), 1–24.

[27] M. A. Taylor, B. A. Wingate and R. E. Vincent, An algorithm for computing Fekete points in the triangle, SIAM J. Numer. Anal., 38(5) (2000), 1707–1720.

[28] A. Toselli and O. Widlund, Domain decomposition methods-algorithms and theory, volume 34 of Springer Series in Computational Mathematics, Germany: Springer-Verlag, 2005.

[29] X. Tu, Three-level BDDC in two dimensions, Int. J. Numer. Methods Eng., 69(1) (2007), 33–59.