Support-Graph Preconditioners*
CM '98 Abstract, Speaker: Sivan Toledo

Marshall Bern John R. Gilbert Bruce Hendrickson Nhat Nguyen
Sivan Toledo

January 16, 1998

1 Introduction

This paper presents new applications of a little-known technique for constructing and analyzing preconditioners called *support-graph preconditioning.* The technique was first proposed and used by Pravin Vaidya [4], who described it in a talk in 1991, but did not publish a paper. Vaidya used the technique to design a family of novel preconditioners. Later, Gremban [2] extended the technique and used it to construct another family of preconditioners. This paper explains the technique, extends it further, and uses it to analyze two classes of *known* preconditioners for model problems. Specifically, we use the extended technique to analyze certain modified-incomplete-Cholesky (MICC) preconditioners and multilevel-diagonal-scaling (MDS) preconditioners.

Analyzing known preconditioners using the support-graph technique serves several purposes. First, it obviously shows that the technique is more widely applicable than previously known. Second, we feel that the new proofs provide useful insights into these preconditioners; these insights can be used to improve the preconditioners and to guide heuristics for the construction of additional preconditioners. Third, we feel that the proofs are quite elementary and thus very appropriate for inclusion in courses on iterative methods.

The support-graph technique analyzes a preconditioner B for a matrix A by splitting both A and B into $A = A_1 + A_2 + \cdots + A_m$ and $B = B_1 + B_2 + \cdots + B_m$. Proving that $\tau B_i - A_i$ is positive semidefinite[1] for all i shows that $\tau B - A$ is semidefinite and hence that the largest finite generalized eigenvalue of the matrix pair (A, B) is bounded by τ. The bound on the smallest generalized eigenvalue is proved by bounding the largest eigenvalue of (B, A) in the same way. The splitting of A and B into simple terms often allows us to reduce a complex problem into many simple problems.

The abstract, as well as the proposed CM '98 talk, consists of three parts besides this introduction. Section 2 explains the theory behind support-graph preconditioning. This includes the theory developed by Vaidya and Gremban, as well as our new extensions. Section 3 presents our new analyses of MICC and MDS preconditioners, as well as some numerical experiments. We present our conclusions in Section 4. Appendix A includes a brief presentation of Vaidya and Gremban's preconditioners, which will also be described in the talk.

*Gilbert and Toledo are supported in part by DARPA contract number DABT63-95-C-0087 and by NSF contract number ASC-96-26298. Hendrickson was funded by the Applied Mathematical Sciences program, U.S. Department of Energy, Office of Energy Research and works at Sandia National Laboratories, operated for the U.S. DOE under contract number DE-AL04-94AL8500.

[0] Marshall Bern and John R. Gilbert are with the Xerox Palo Alto Research Center. Bruce Hendrickson is with Sandia National Laboratory. Nhat Nguyen is with Stanford University. Sivan Toledo is with the Xerox Palo Alto Research Center, on leave from Tel Aviv University.

[1] From here on we use the term *semidefinite* to denote positive semidefinite matrices, unless explicitly noted otherwise.

2 Support-Graph Theory

This section sketches support-graph theory. Proofs are omitted from this abstract. The main goal of the theory is to bound the extreme finite generalized eigenvalues of a matrix pair (A, B), where both A and B are semidefinite. The ratio of the extreme eigenvalues is the condition number of the preconditioned linear system (where B is used as a preconditioner for A), which in turn can be used to estimate the convergence rate of iterative solvers.

Definition The number λ is a finite generalized eigenvalue of the matrix pencil (A, B) if there exists a vector x such that $Ax = \lambda Bx$ and λ is unique. We denote the set of finite generalized eigenvalues by $\lambda(A, B)$.

The main tool that we use to upper bound $\lambda(A, B)$ is the so-called *support* of (A, B), denoted by $\sigma(A, B)$, which is the smallest number τ such that $\tau B - A$ is semidefinite, where B is a semidefinite matrix. The support of a pencil bounds its eigenvalues. A more general version of this lemma can be found in Axelsson [1, Theorem 10.1].

Lemma 1 (Support Lemma [2]) *If $\lambda \in \lambda(A, B)$ where B is semidefinite and* $\mathrm{null}(A) \subseteq \mathrm{null}(B)$*, then*

$$\lambda \leq \sigma(A, B) = \min\{\tau : \tau B - A \textit{ is semidefinite}\} .$$

We use the Support Lemma to prove an upper bound τ on $\lambda(A, B)$ by proving that $\tau B - A$ is semidefinite. Much of the rest of the theory consists of tools to prove that a matrix is semidefinite. One way to prove that a matrix is semidefinite is to split it into a sum of matrices and prove that each term is semidefinite.

Lemma 2 (Splitting Lemma [2]) *Let $Q = Q_1 + Q_2 + \cdots + Q_m$ be a square matrix such that $Q_1, Q_2, \ldots, Q_m$ are all semidefinite. Then Q is semidefinite.*

We first use the Splitting Lemma to reduce the problem of preconditioning symmetric diagonally-dominant matrices to the problem of preconditioning symmetric matrices with zero row sums.

Lemma 3 *Let A be a symmetric diagonally-dominant matrix and let A' be a matrix with the same offdiagonal entries but with zero row sums. Let B' be a preconditioner for A' such that both $\beta B' - A'$ and $\alpha A' - B'$ are semidefinite and $\alpha, \beta \geq 1$. Let $B = B' + A - A'$ be a preconditioner for A (B has the same offdiagonal entries as B' and the same row sums as A). Then $\beta B - A$ and $\alpha A - B$ are semidefinite.*

Using this lemma, we assume from here on that A and B have zero row sums.

Vaidya and Gremban split $\tau B - A$ in a special way to prove that it is semidefinite. Let G_A be the weighted graph underlying $-A$ but with no self loops, and G_B the graph underlying $-B$. Since both A and B have zero row sums, the graphs determine the matrices exactly, since a diagonal element is determined by the sum of the offdiagonal elements in its row. If the offdiagonal elements of A and B are all negative, then the edge weights in G_A and G_B are positive. Vaidya and Gremban interpret such graphs as resistive networks where the edge weight is the conductance of a resistor. They split $\tau B - A$ into $(\tau B_1 - A_1) + (\tau B_2 - A_2) + \cdots + (\tau B_m - A_m)$ such that each A_i corresponds to exactly one edge in G_A, and each B_i corresponds to a simple path in G_B. Both the A_i's and the B_i's have nonpositive offdiagonals and zero row sums. Each A_i represents the entire weight of one edge, and each corresponding B_i represents a path that can contain fractions of edge weights. The endpoints of the path represented by B_i are the endpoints of the edge represented by A_i. An example is shown in figure 1. Both Vaidya and Gremban use the following lemma to show that each term $\tau B_i - A_i$ is semidefinite.

Lemma 4 (Congestion-Dilation Lemma) *Let*

$$A = \begin{pmatrix} a & 0 & \cdots & 0 & -a \\ 0 & 0 & & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & 0 & 0 \\ -a & 0 & \cdots & 0 & a \end{pmatrix} .$$

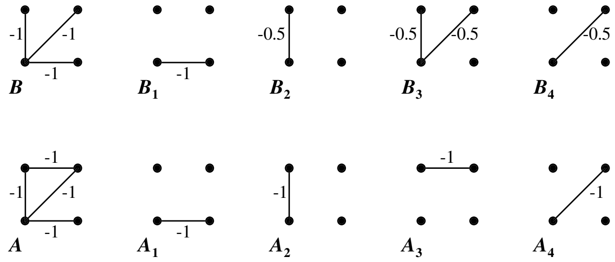

Figure 1: A graph representation of a splitting of $A = A_1 + \cdots + A_4$ and $B = B_1 + \cdots + B_4$ such that each A_i is represents a single edge and each B_i is a path that supports the edge A_i. This splitting proves that $\lambda(A,B) \le \sigma(A,B) \le 4$ since the worst congestion-dilation product is $2 \cdot 2 = 4$.

be a $(k+1)$-by-$(k+1)$ matrix with $a > 0$ and let

$$B = \begin{pmatrix} d_1 & -b_1 & & & \\ -b_1 & d_2 & -b_2 & & \\ & & \cdots & & \\ & & -b_{k-1} & d_k & -b_k \\ & & & -b_k & d_{k+1} \end{pmatrix} .$$

Then $(k \cdot a/\min(b_i))B - A$ is semidefinite.

The combinatorial interpretation of the Congestion-Dilation Lemma is that a/b is the *congestion* of the edge represented by A_i in the path represented by B_i, and k is the *dilation* of the edge, or the simply the length of the path.

The Support, Splitting, and Congestion-Dilation Lemmas are the only linear-algebra tools that Vaidya uses in his construction. Gremban, however, introduced a few more linear-algebra tools into the combinatorial preconditioning theory. The first lemma, stated under a stronger hypothesis by Gremban [2, Lemma 4.9], shows that $\lambda(A,B)$ is invariant under non-singular transformations that are applied to both A and B.

Lemma 5 *Let G and H be nonsingular matrices. Then*

$$\lambda(A,B) = \lambda(GAH, GBH) .$$

The second lemma is a stronger version of the Support Lemma. Gremban does not use this lemma but a similar one that is tailored more precisely to his technique. We state (and prove in the full version of this abstract) the more general case.

Lemma 6 *If $\lambda \in \lambda(A,B)$ where $A \neq 0$ and B are symmetric semidefinite, then*

$$\begin{aligned} \lambda &\le \bar{\sigma}(A,B) \\ &= \min\{\tau : x^T(\tau B - A)x \ge 0 \textit{ for all } x \textit{ such that} \\ &\qquad Ax \neq 0 \textit{ and } Bx \neq 0\} . \end{aligned}$$

We call $\bar{\sigma}(A,B)$ the generalized support *of (A,B).*

The importance of these two lemmas stem from the fact that they allow us to reduce the case where

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} B_{11} & 0 \\ 0 & 0 \end{pmatrix} \tag{1}$$

to a simpler case where

$$A' = \begin{pmatrix} A'_{11} & 0 \\ 0 & A'_{22} \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} B_{11} & 0 \\ 0 & 0 \end{pmatrix} \tag{2}$$

by eliminating the $(2,2)$ block of A. The eigenvalues of the matrix pair are maintained under this transformation, but the case (2) is easier to analyze because the null space of B is contained in a subspace that is invariant under $\tau B - A$.

Support Graph Theory: Extensions

We now describe two *new* tools that extend the support-graph theory that Vaidya and Gremban developed.

The first technique allows us to analyze support graphs that are larger than A, like Gremban's preconditioners, but without performing a complete elimination of the extra vertices that are in B but not in A. Our technique relies on the following two lemmas. The first lemma is a version of Lemma 5 but for support numbers rather than eigenvalues.

Lemma 7 *Let G be a nonsingular matrix. Then $\bar{\sigma}(A,B) = \bar{\sigma}(G^TAG, G^TBG)$.*

The second lemma shows that we can subtract certain semidefinite matrices from A without increasing $\bar{\sigma}(A,B)$.

Lemma 8 (Shifting Lemma) *Let C be a semidefinite matrix such that* $\text{null}(A) \subseteq \text{null}(C)$. *Then*

$$\bar{\sigma}(A - C, B) \leq \bar{\sigma}(A, B) \ .$$

These lemmas allows us to reduce (1) to (2) in a series of phases. In each phase, we perform one or more steps of Gaussian elimination on A, followed by a subtraction of a negative semidefinite matrix from A (addition of a positive semidefinite matrix). After all of these steps are complete, we prove a bound on the generalized support of the resulting matrices. These lemmas allow us to then retrace the steps that we have taken, performing transformations that reverse the effects of elimination and subtraction steps, while preserving the support bound. Thus, the bound that we prove on the matrices after this series of transformations is also a bound on the original matrix and preconditioner.

Positive offdiagonal entries in B require a modification to the splitting strategy. Recall that the "canonical" use of the Splitting Lemma is to split $\tau B - A$ into $(\tau B_1 - A_1) + \cdots + (\tau B_m - A_m)$, where the B_i's are semidefinite. When the Congestion-Dilation Lemma is used, the B_i's are usually paths of negative edges, and the A_i's are negative edges, all with zero row sums (positive row sums can be handled separately as shown in Lemma 3). When B has positive offdiagonal entries, this strategy must be modified. One way to prove an upper bound on $\sigma(A, B)$ in this case using the congestion-dilation framework is to split $\tau B - A$ into

$$\begin{aligned} \tau B - A &= (\tau B_1 - A_1) + \cdots + (\tau B_m - A_m) \\ &\quad + (\tau B_{m+1} + \tau B_{m+k+1}) + \cdots \\ &\quad \cdots + (\tau B_{m+k} + \tau B_{m+2k}) \end{aligned}$$

and prove that each term is semidefinite, where B_{m+k+1} through B_{m+2k} are submatrices representing a single positive edge each, and B_1 through B_{m+k} are paths of negative edges. Thus, B_1 through B_m support edges of A, while B_{m+1} through B_{m+k} *support the positive edges of B*. Note that in this case, increasing τ does not necessarily make all the terms semidefinite. Indeed, if $\tau B_{m+j} + \tau B_{m+k+j}$ is indefinite or negative definite, then it remains so for all $\tau \geq 0$. In other words, the positive edges of B must be routed by paths with support at most 1.

A special case of this technique is applicable for preconditioners that are produced by incomplete factorizations. These preconditioners can be represented as $B = A - R$, where R contains the entries that are dropped during the factorization and any diagonal modification that is performed. The following lemma simplifies the analysis in this special but important case.

Lemma 9 *Let $B = A - R$ such that R is semidefinite. If $\sigma(R, A) = \tau' < 1$, then $\kappa(B^{-1}A) \leq 1/(1-\tau')$.*

In essence, the lemma allows us to bound $\lambda(A, B)$ by bounding the support of R in A.

3 Analysis of MICC and MDS Preconditioners

This section describes our new applications of support-graph theory.

MICC Preconditioners

We begin by describing our analysis of preconditioners based on modified incomplete factorizations.

Let $B = LL^T$ be a level-0 modified incomplete factorization of an M-matrix A. The incomplete factor L has the same nonzero structure as the lower triangle of A, and B has the same row sums as A. We can write $B = A - R$, where R consists of the fill elements that are dropped during the factorization plus the diagonal modification that is performed in order to maintain the row sums. Since the elements that are dropped are always negative and since A and B have the same row sums, R is an M-matrix with zero row sums.

We now analyze the modified incomplete Cholesky factorization for a 2-dimensional model problem, a Laplace equation with Neumann boundary conditions on a regular grid. The following theorem is a well-known result; to the best of our knowledge, the proof, which is omitted from this abstract, is novel.

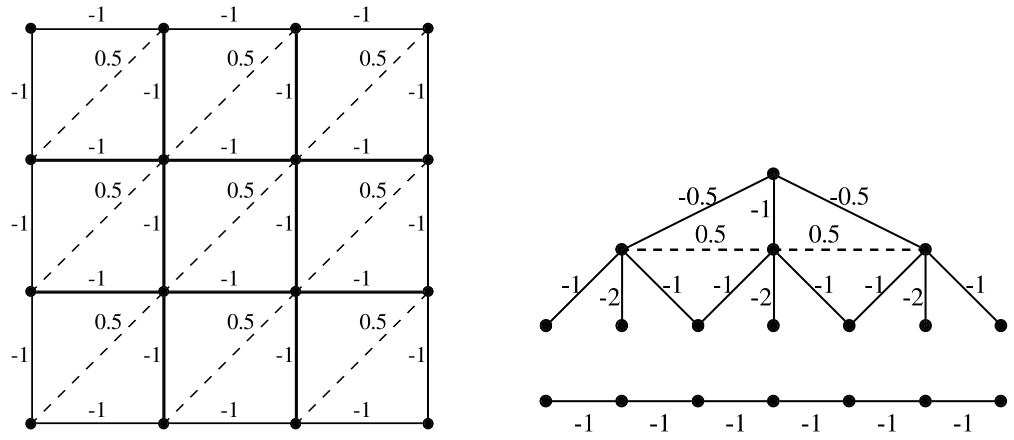

Figure 2: Left: a modified incomplete-Cholesky preconditioner B for a 2D Poisson problem. The preconditioner consists of the entries in the matrix A (negative values, solid lines) and of positive valued edges (dashes) that cancel fill. Right: a 1D Poisson problem (bottom) and its multilevel preconditioner (top). Positive edges in the preconditioner (dashed) cancel fill in a bottom up factorization of the preconditioner.

We have recently learned that Guattery [3] has performed a somewhat similar analysis of unmodified incomplete factorizations.

Theorem 1 *Let A represent a model Neumann problem discretized on a $\sqrt{n}$-by-$\sqrt{n}$ grid. Let B be a modified incomplete-Cholesky factorization of A with no fill, as shown in Figure 2. Then $\kappa(B^{-1}A) \leq 2\sqrt{n} - 2$.*

Proof: We only sketch the proof here. The main idea is to use Lemma 9 to bound the condition number. Lemma 9 requires us to prove a bound on the support of R (a matrix consisting of the positive edges in the preconditioner) in A. We prove this bound using a congestion-dilation argument that routes each positive edge in two length-2 paths in A. $\square$

A generalization of this result shows that the condition number of a modified incomplete-Cholesky preconditioner for a 3D Poisson problem is $O(n^{2/3})$ or better.

MDS Preconditioners

We next turn our attention to MDS preconditioners.

We sketch here the algorithm that we use to construct these preconditioners. We decompose the vertices of G_A into "subdomains" in such a way that both endpoints of each edge in G_A belong to some subdomain, possibly more than one. We associate a new vertex with each subdomain and add edges in G_B that connect the new vertex to all the vertices in the subdomain. We call these vertices *subdomain vertices.* Each edge of G_A is routed in a path of length 2 going from one of its endpoints to the new subdomain vertex and back to the other endpoint. An edge whose endpoints belong to more than one subdomain is routed in all of them. The relative weight of the edges that connect a subdomain vertex in to the vertices in the subdomain is chosen so that the congestion-dilation products for all the edges in the subdomain are identical. We then scale these edges so that when the subdomain vertex is eliminated, the G_A edges in the subdomain support the fill edges that result from the elimination with support at most 1.

Next, we add positive edges between subdomain vertices. To determine these edges, we eliminate all the original vertices in the graph. This elimination creates fill edges that are negative if A is an M-matrix. We add positive edges that exactly cancel the fill. These positive edges form a "coarse grid" that also needs to be supported. We therefore decompose the subdomain vertices into new subdomains, and add new subdomain vertices and edges to them so that they support the positive "coarse grid" edges.

The end result of this process is a hierarchical structure that can be factored bottom up with no fill. In fact, the factors of G_B are sparser than G_B, just as in the case of the modified incomplete-Cholesky factorization.

Figure 2 shows the result of this construction for a 1D model Poisson problem. The next theorem shows that the condition number of the preconditioned system in this 1D case is $O(\log n)$.

Theorem 2 *Let A be a tridiagonal matrix of size on $n = 2^k - 1$ with diagonal elements $A_{i,i} = 2$ and offdiagonal elements $A_{i,i+1} = A_{i+1,i} = -1$, and let T be the corresponding hierarchical support graph, as shown in Figure 2. Then $\lambda(A,T) \in [1,k]$.*

Proof: Again, we only sketch the proof here. We upper bound $\sigma(A,T)$ by routing both the edges of A and the positive edges of T in the negative edges of T. As in Lemma 9, there are two length-2 paths that support every such edge, and we use both. We set up a recurrence relation that determines how much of each edge can be routed in each path. The solution of this recurrence shows that the $1/k$ of each edge in G_A can be routed through T, so $\sigma(A,T) \leq k$. To bound $\bar{\sigma}(T,A)$, we use Lemmas 7 and 8. These lemmas allow us to perform an elimination step on the top level subdomain vertices and to remove both the resulting fill edges and the original positive edges in T while maintaining a support bound of 1. Thus $\lambda(A,T) \in [1,k]$. $\square$

Although we have not been able to prove similar logarithmic or polylogarithmic condition-number bounds on MDS preconditioners in two dimensions, numerical experiments show that their condition numbers grow quite slowly. For example, we have constructed a family of preconditioners for a model problem discretized on a regular triangular 2D mesh. Our experiments show that the number of iterations required to solve the problem grows slowly as a function of the number of mesh points. Our prototype implementation (in Matlab) works for M-matrices with arbitrary sparsity patterns and we are conducting additional numerical experiments.

n	No Precond.	ICC(0)	MDS
15	8	6	10
45	20	8	14
153	36	12	15
561	64	19	15
2145	117	33	17
4753	154	45	20

Table 1: The number of iterations it takes preconditioned conjugate gradient to converge on a model Poisson problem discretized on an n-point regular triangular grid. The table shows the number of iterations without preconditioning, with a ICC(0) preconditioner, and with our MDS preconditioner.

4 Conclusions

This paper describes new applications of a preconditioning technique called support-graph preconditioning. Our new results extend the theory of support-graph preconditioners and demonstrate that it be applied to the analysis of well-known preconditioners. The insights that we have gained from this analysis guide us in the construction of new preconditioners whose condition numbers cannot be fully analyzed by our techniques.

We believe that support-graph theory is a significant contribution to the field of linear-system preconditioning and that it has not gained the attention it deserves. The significance of the theory stems from the fact that it bring new tools to this field. The new tools shed new light on existing preconditioners and enable the discovery of new ones. Vaidya's preconditioners, which are described in the full paper, are perhaps the best example of the novelty of the theory. These preconditioners are efficient, novel, and easy to analyze. They are asymptotically faster than modified incomplete-Cholesky preconditioners for 2D model problems and they deliver the same performance guarantee to any planar M-matrix.

References

[1] Owe Axelsson. *Iterative Solution Methods*. Cambridge University Press, 1994.

[2] Keith D. Gremban. *Combinatorial Preconditioners for Sparse, Symmetric, Diagonally Dominant Linear Systems.* PhD thesis, School of Computer Science, Carnegie Mellon University, October 1996. Technical Report CMU-CS-96-123.

[3] Stephen Guattery. Graph embedding techniques for bounding condition numbers of incomplete factor preconditioners. Technical Report ICASE Report 97-47, NASA Langley Research Center, 1997.

[4] Pravin M. Vaidya. Solving linear equations with symmetric diagonally dominant matrices by constructing good preconditioners. An unpublished manuscript. A talk based on the manuscript was presented at the IMA Workshop on Graph Theory and Sparse Matrix Computation, October 1991, Minneapolis.

A Vaidya and Gremban's Preconditioners

This section briefly describes the preconditioners that Vaidya [4] and Gremban [2] proposed. We describe the preconditioners for the case where A is an M-matrix, although Gremban and some of Vaidya's preconditioners can be extended to symmetric diagonally-dominant matrices.

Given an n-by-n matrix A with m nonzeros, Vaidya's method [4] starts by finding a maximum spanning tree in G_A (a spanning tree that maximizes the total weight of the edges in it). Vaidya's first preconditioner simply uses this tree as the preconditioner for A. The preconditioning matrix B thus contains a subset of the offdiagonal nonzeros of A and has the same row sums. This tree is easy to find in at most $O(m \log m)$ work. Factoring the tree and using the factor in a preconditioning step both cost only $O(n)$. Vaidya shows that if A has at most a fixed number of nonzeros per row, $\lambda(A, B) \in [1, O(n^2)]$.

Vaidya then proposes to improve the preconditioner by adding more edges to the tree in a certain way. Adding more edges makes the preconditioner more expensive to factor, but also reduces the condition number of the preconditioned system. Therefore, how many edges are added depends on a tradeoff between the preconditioner-factorization cost and the cost of the iterations. If G_A does not have any special structure, then the optimal tradeoff leads to an iterative solution that uses $O(n^{1.75})$ work (that is, the condition number is $O(n^{1.5})$ and the cost of constructing and factoring the preconditioner is about the same as the cost of the iterations). If G_A is planar, The total cost is $O(n^{1.2})$.

Gremban proposes a multilevel preconditioner that is based on a hierarchical partitioning of G_A. Gremban's method constructs a graph G'_B with more vertices then G_A, so he also augments G_A with additional disconnected vertices, so that both graphs use the same vertex sets. In matrix terms, the construction embeds A an the leading block of a larger zero matrix.

Gremban's preconditioner G_B has one vertex for each subgraph in the hierarchical partitioning of G_A, starting from a single vertex that represents the entire G_A and ending with n vertices that represent single vertices in G_A. A vertex in the preconditioner that represents a subgraph S of G_A is connected to the vertices that represent the subgraphs into which S is partitioned and to the vertex that represents the subgraph that contains S as one of its parts. The weight of the edge that connects S to the subgraph that contains it is the total weight of the edges that separate S from the rest of G_A. In short, structure of G'_B represents the hierarchical decomposition of G_A and the edge weight of G'_B represent the weights of the "boundaries" of subgraphs in the partition. Since $G(B')$ is guaranteed to be a tree with diameter $O(\log n)$, factoring B' and applying the factors in every iteration requires only $O(n)$ work and $O(\log n)$ parallel steps. The cost of computing $G(B')$ depends on the graph-partitioning algorithm that is used and may be substantial in practice.

Gremban shows that for model Poisson problems discretized on regular rectangular meshes in any fixed dimension, the condition number of the preconditioned system is $O(\sqrt[d]{n} \log n)$. We omit the details from this abstract.