Overlapping Domain Decomposition Preconditioners for the Generalized Davidson Method for the Eigenvalue Problem *

Andreas Stathopoulos †, Youcef Saad ‡, Charlotte. F. Fischer†

February 15, 1994

1 Introduction

The solution of the large, sparse, symmetric eigenvalue problem, $Ax = \lambda x$, is central to many scientific applications. Among many iterative methods that attempt to solve this problem, the Lanczos and the Generalized Davidson (GD) are the most widely used methods. The Lanczos method builds an orthogonal basis for the Krylov subspace, from which the required eigenvectors are approximated through a Rayleigh-Ritz procedure. Each Lanczos iteration is economical to compute but the number of iterations may grow significantly for difficult problems. The GD method can be considered a preconditioned version of Lanczos [4, 2]. In each step the Rayleigh-Ritz procedure is solved and explicit orthogonalization of the preconditioned residual $((M - \lambda I)^{-1}(A - \lambda I)x)$ is performed. Therefore, the GD method attempts to improve convergence and robustness at the expense of a more complicated step.

Variations of the Schwarz domain decomposition algorithms are extensively used for solving linear systems arising from Partial Differential Equations [3]. They solve restrictions of the problem on different subdomains independently, and then integrate the partial solutions. For this reason they have become popular on parallel computers. Solving the subdomain problems is expensive in large, sparse matrices and therefore Schwarz algorithms have been alternatively used as powerful preconditioners [3, 1]. Their power is further enchanced by the fact that they allow for domain overlapping. Increased domain overlapping as well as the use of a coarse grid solver compensates for the convergence deterioration with the number of subdomains. When no overlapping is used, the popular additive and multiplicative Schwarz algorithms are the usual block Jacobi and Gauss-Seidel ones.

Schwarz domain decomposition algorithms can be used to precondition the algebraic eigenvalue problem as well. The additive and multiplicative Schwarz methods have been applied on the GD method for solving the equation $(M - \lambda I)\delta = (A - \lambda I)x$ in a multicomputer environment. However, in this environment the subdomain overlapping causes some eigenvalue-specific problems. This paper presents the changes that are necessary to solve these problems. Some preliminary experimental results are also given.

*This work was supported by National Science Foundation under grant numbers ASC-9005687 and DMR-9217287, and by AHPCRC (University of Minnesota) under Army Research Office grant number DAAL03-89-C-0038.

†Computer Science Department, Vanderbilt University, Nashville, TN.

‡Computer Science Department, University of Minnesota

2 Problems from Subdomain Overlapping on GD

Assuming that the smallest eigenpair of $Ax = \lambda x$ is sought and B is an initial basis of an approximating subspace, a conceptual description of the GD algorithm follows[7, 6]:

GD Algorithm
Step 1. Compute projection $S = B^T AB$.
Step 2. Solve $Sc = \lambda c$
Step 3. Compute the residual $R = (ABc - \lambda Bc)$.
Step 4. Precondition: $R' = (M - \lambda I)^{-1} R$.
Step 5. Orthogonalize: $b' = (I - BB^T)R'$.
Step 6. Add normalized vector $b'/\|b'\|$ to B and repeat from 1.

The distribution of the algebraic problem onto the multicomputer is similar to the one in Sobolev spaces. Each processor holds a number of rows (nodes) of the matrix and the corresponding components of the eigenvectors and work arrays. Usually, a subdomain is associated with the rows on each processor and overlapping subdomains hold a number of the same rows.

On multicomputers, subdomain overlapping alters the dot product. The reason for this is that the overlapped regions of vectors contribute to the dot-product more than once. More specifically, if D is the overlap diagonal matrix, where $D_{i,i}$ is the number of processors on which row i appears, for any distributed vectors x, y the altered product $(.,.)_{ovl}$ is given by:

$$(x, y)_{ovl} = (x, y)_D = (Dx, y),$$

In linear systems of equations this is not a problem by itself, since the Galerkin condition does not depend on the dot-product. However, the Rayleigh-Ritz procedure is sensitive to the dot-product:

$$x = Bc, \ (Ax - \lambda x, B)_D = 0 \ \Rightarrow \ B^T DABc - \lambda B^T DBc = 0.$$

A second problem is that the $(.,.)_{ovl}$ dot-product describes the correct change only when the overlapping regions are the same in different processors. However, preconditioning operates on different subdomains with different sections of A, and the resulting overlapped regions in different processors do not coincide. Thus, the dot-product does not have a succinct formula as the above one and simple scaling cannot be used.

3 Proposed Solutions

• The problem of non coinciding overlapping regions in different processors, needs to be dealt with first, because otherwise the effect of the dot-product is not easily recordable. The problem can be faced by introducing a communication step after preconditioning. The overlapped regions of the preconditioned vector are communicated and weighed (averaged by D^{-1}) over all the processors. Since the size of overlaps is usually much smaller than the dimension of A this should not present a bottleneck in the algorithm. After this step, all overlapped vector components have identical values on the processors and the dot-product can be safely assumed to be $(.,.)_{ovl}$.
• There are two ways to deal with the change in the dot-product. The simplest (and most intuitive) one is to correct the effect of the dot-product by scaling one of the vectors back with D^{-1}. The dot-product defined by $(x, D^{-1}y)_{ovl} = (x, DD^{-1}y) = (x, y)$ is equal to the correct (non-overlapping) one. This obvious solution has two disadvantages: the GD library routine must be modified accordingly for the new dot-product, and requires $k+2$ scalings in each iteration, where k is the size of the basis.

• A different approach to the same problem is to consider the scaled matrix $A_s = D^{-\frac{1}{2}} A D^{\frac{1}{2}}$ and the scaled initial basis $B_s = D^{-\frac{1}{2}} B$. If the GD algorithm is applied to these scaled matrices with $(.,.)_{ovl}$ dot-product and with the above averaging after preconditioning, it gives the correct eigenvalue and the scaled eigenvector. To show this claim the following are necessary.

Let the rows of the matrix A span $\Re^N$. Let also the rows of the matrix local to subdomain i span the subspace L_i. If I_i is the orthocanonical basis of L_i, $P = I_i I_i^T$ is an orthogonal projector onto L_i. The *section* of A on L_i is defined as: $A_i = PAP = I_i I_i^T A I_i I_i^T$. Although A_i is not invertible, its restriction to L_i can be inverted. Let A_i^{-1} be defined as:

$$A_i^{-1} \equiv I_i (I_i^T A I_i)^{-1} I_i^T.$$

With this definition, the algebraic formulation of the additive and multiplicative Schwarz preconditioners is:

$$\begin{aligned} M^{-1} &= A_1^{-1} + \ldots + A_q^{-1} && \text{(additive)}, \\ M^{-1} &= (I - (I - A_1^{-1} A) \cdots (I - A_q^{-1} A)) A^{-1} && \text{(multiplicative)}. \end{aligned}$$

Lemma 1 *The inverted sections of A_s are the scaled inverted sections of A, i.e.,*

$$(A_s)_i^{-1} = D^{-\frac{1}{2}} A_i^{-1} D^{\frac{1}{2}}.$$

Proof.

$$\begin{aligned} (A_s)_i^{-1} &= I_i (I_i^T A_s I_i)^{-1} I_i^T \\ &= I_i (I_i^T D^{-\frac{1}{2}} A D^{\frac{1}{2}} I_i)^{-1} I_i^T \\ &= I_i (I_i^T I_i I_i^T D^{-\frac{1}{2}} A D^{\frac{1}{2}} I_i I_i^T I_i)^{-1} I_i^T \\ &= I_i ((I_i^T D^{-\frac{1}{2}} I_i)(I_i^T A I_i)(I_i^T D^{\frac{1}{2}} I_i))^{-1} I_i \\ &= I_i (I_i^T D^{\frac{1}{2}} I_i)^{-1} (I_i^T A I_i)^{-1} (I_i^T D^{-\frac{1}{2}} I_i)^{-1} I_i^T && (1) \\ &= I_i I_i^T D^{-\frac{1}{2}} I_i (I_i^T A I_i)^{-1} I_i^T D^{\frac{1}{2}} I_i I_i^T && (2) \\ &= D^{-\frac{1}{2}} I_i (I_i^T A I_i)^{-1} I_i^T D^{\frac{1}{2}} \\ &= D^{-\frac{1}{2}} A_i^{-1} D^{\frac{1}{2}}. \end{aligned}$$

The transition from eq. (1) to eq. (2) is justified because D is a diagonal matrix and I_i part of the identity matrix. □

As a consequence of the above Lemma, the additive and multiplicative Schwarz preconditioners derived from the scaled A_s are simply the scaled preconditioners of A.

Proposition 1 *If M_s is the additive or multiplicative Schwarz preconditioner of A_s, and M is the corresponding preconditioner of A, then*

$$M_s^{-1} = D^{-\frac{1}{2}} M^{-1} D^{\frac{1}{2}}.$$

Proof. For the additive case and from Lemma 1:
$M_s^{-1} = \sum_{i=1}^q (A_s)_i^{-1} = \sum_{i=1}^q D^{-\frac{1}{2}} A_i^{-1} D^{\frac{1}{2}} = D^{-\frac{1}{2}} (\sum_{i=1}^q A_i^{-1}) D^{\frac{1}{2}} = D^{-\frac{1}{2}} M^{-1} D^{\frac{1}{2}}$.
For the multiplicative case:
$M_s^{-1} = (I - \prod_{i=1}^q (I - (A_s)_i^{-1} A_s)) A_s^{-1} = (I - \prod_{i=1}^q (I - D^{-\frac{1}{2}} A_i^{-1} A D^{\frac{1}{2}})) D^{-\frac{1}{2}} A^{-1} D^{\frac{1}{2}} =$
$= D^{-\frac{1}{2}} (I - \prod_{i=1}^q (I - A_i^{-1} A)) A^{-1} D^{\frac{1}{2}} = D^{-\frac{1}{2}} M^{-1} D^{\frac{1}{2}}$. □

With the above results it is easy to show that each step of the GD procedure, applied to A_s with starting basis B_s, computes the correct eigenvalues of A and their corresponding eigenvectors scaled by $D^{-\frac{1}{2}}$. Assume $\|B\| = 1$ for the original (not the overlapped) norm.

Step 1 The computed projection matrix is $S = (B_s, A_sB_s)_{ovl} = B^TD^{-\frac{1}{2}}DD^{-\frac{1}{2}}AD^{\frac{1}{2}}D^{-\frac{1}{2}}B =$ $= B^TAB$, i.e., the same with the correct projection of the unscaled matrix.

Step 2 This depends only on S and therefore is not affected.

Step 3 The residual $R_s = (A_s - \lambda I)B_sc = D^{-\frac{1}{2}}R$, where R is the residual of the unscaled method.

Step 4 According to Proposition 1, $R'_s = M_s^{-1}R_s = D^{-\frac{1}{2}}M^{-1}D^{\frac{1}{2}}D^{-\frac{1}{2}}R = D^{-\frac{1}{2}}R'$, where R' is the preconditioned residual of the unscaled method.

Step 4′ Both the scaled and unscaled versions, introduce different values on the overlapping regions of R' and R'_s. Thus, they both require a global summation and scaling: $R'_s \leftarrow D^{-1}(\sum_{j=1}^{q} R_s'^{(j)})$, $R' \leftarrow D^{-1}(\sum_{j=1}^{q} R'^{(j)})$, and the relation $R'_s = D^{-\frac{1}{2}}R'$ still holds.

Step 5 The orthogonalized vector $b'_s = R'_s - B_s(B_s, R'_s)_{ovl} = D^{-\frac{1}{2}}(R' - BB^TR') = D^{-\frac{1}{2}}b'$, where b' is the correctly orthogonalized vector of the unscaled version.

Step 6 The computed norm of b'_s is $\|b'_s\|^2 = (b'_s, b'_s)_{ovl} = b'^tD^{-\frac{1}{2}}DD^{\frac{1}{2}}b' = b'^Tb'$. Therefore the norm computed is the correct norm of the new basis vector b' of the unscaled case. The algorithm can now repeat from Step 1.

Several applications scale the matrix A before the GD is applied. In these cases the scaling $D^{-\frac{1}{2}}AD^{\frac{1}{2}}$ can be done at no additional cost. M need not be explicitly scaled since it is extracted from A. Scaling the matrix A may also be beneficial in cases of sparse matrices with large overlap where the GD requires many iterations.

Another way of exploiting the above alternative algorithm is not to scale the matrix A but to record the effects of the scaled one in the matrix-vector multiplication and preconditioning routines. Since $A_s = D^{-\frac{1}{2}}AD^{\frac{1}{2}}$ and $M_s^{-1} = D^{-\frac{1}{2}}M^{-1}D^{\frac{1}{2}}$, an operation of any of these two matrices (say F) to a vector from the above algorithm can be formed as $D^{-\frac{1}{2}}(F(D^{\frac{1}{2}}b)))$. This is equivalent to scaling the vector by $D^{\frac{1}{2}}$ before the operation with F and by $D^{-\frac{1}{2}}$ after the operation, i.e., two scalings per operation. Moreover, the scaling after the preconditioning (by $D^{-\frac{1}{2}}$) can be combined with the scaling from averaging (by D^{-1}) into one scaling by $D^{-\frac{3}{2}}$. Thus, this approach costs 3 scalings per GD iteration and is much cheaper than the initial approach.

4 Preliminary Results and Conclusions

The use of overlapping domains in additive and multiplicative Schwarz preconditioners for the GD is demonstrated with some preliminary results on Tables 1 and 2. The codes used are described in [1, 5, 7], and the experiments are carried out on a PVM 4-node multiprocessor. No underlying grid is considered and the partitioning of the matrix is performed through automatic domain decomposition tools. The subdomain problems are solved with ILU(0).

On both tables the overlap is denoted by two numbers; the number of breadth first search levels that a domain is expanded, and the maximum number of additional nodes per level allowed. The number of GD iterations is reported for the additive (Add) and multiplicative (Mult) preconditioners. On Table 1, the matrix used is BCSSTK07 from the Harwell-Boeing collection. Its dimension is 420 and it has poor eigenvalue separation for the lowest part of the spectrum. The lowest eigenpair is sought for. Two steps of the corresponding Schwarz method are applied in each iteration. For the Add method the overlap (10,2) improves the convergence significantly, while (8,6) takes more than three times the steps. Improvements for the Mult method are more consistent with larger overlaps.

On Table 2, the matrix used comes from atomic structure calculations [7]. Its dimension is 748 and it has fairly good eigenvalue separation for the lower spectrum. The third lowest eigenpair is sought for. Two steps of the additive preconditioner (five for the multiplicative) are

Overlap Levels	0	2	10	4	8
Ovl / level	0	2	2	2	6
Iterations	454	402	360	>700	>1000

Overlap Levels	0	4	10	4	8
Ovl / level	0	2	2	6	6
Iterations	233	258	189	153	154

Table 1: BCSSTK07: Iterations for Additive (left) and Multiplicative (right) preconditioned GD

applied in each iteration. Similarly to the previous case, better and more consistent convergence improvements are observed for the multiplicative algorithm.

Overlap Levels	0	4	10	8	10
Ovl / level	0	2	2	6	6
Iterations	31	31	32	28	28

Overlap Levels	0	4	10	8	10
Ovl / level	0	2	2	6	6
Iterations	25	24	21	21	24

Table 2: Matrix 748: Iterations for Additive (left) and Multiplicative (right) preconditioned GD

These preliminary experiments show that similarly to algebraic linear systems an efficient partition of the algebraic eigenvalue problem requires knowledge of the underlying structure or the use of automatic domain decomposition tools.

Acknowledgement: The authors would like to thank Todd Goehring for the partitioning routines.

References

[1] Xiao-Chuan Cai and Y. Saad, *Overlapping Domain Decomposition Algorithms for General Sparse Matrices,* Preprint 93-027, Army High Performance Computing Research Center, University of Minnesota, 1993.

[2] E.R. Davidson, *The Iterative Calculation of a Few of the Lowest Eigenvalues and Corresponding Eigenvectors of Large Real-Symmetric Matrices*, J. Comput. Phys. 17 (1975) 87.

[3] M. Dryja, and O.B. Widlund, *Towards a unified theory of domain decomposition algorithms for elliptic problems,* in Third International Symposium on Domain Decomposition Methods for PDEs, eds. T.F.Chan et al. (SIAM, Philadelphia 1990).

[4] R.B. Morgan and D.S. Scott, *Generalizations of Davidson's Method for Computing Eigenvalues of Sparse Symmetric Matrices*, SIAM J. Sci. Stat. Comput. 7 (1986) 817.

[5] Y. Saad, *Krylov Subspace Methods in Distributed Computing Environments*, Preprint 92-126, Army High Performance Computing Research Center, University of Minnesota, 1992.

[6] A. Stathopoulos and C. F. Fischer, *Reducing Synchronization on the Parallel Davidson Method for the Large, Sparse, Eigenvalue Problem*, in: Proceed. Supercomputing '93, pp. 172 (ACM Press, Portland 1993).

[7] A. Stathopoulos and C.F. Fischer, *A Davidson Program for Finding a Few Selected Extreme Eigenpairs of a Large, Sparse, Real, Symmetric Matrix*, Comput. Phys. Commun., 79 (1994) 1.