# Support Graph Preconditioners for the Finite Element Method

Vivek Sarin and Meiqiu Wang
Texas A&M University
College Station, TX 77843
USA

## Extended Abstract

In this paper, we present a class of graph-based preconditioners for sparse linear systems arising from the finite element discretization of elliptic partial differential equations. A sparse matrix can be preconditioned by using a subgraph of its weighted adjacency matrix. The subgraph can be viewed as a "support graph" and the corresponding sparse matrix can be used as a preconditioner. The main idea is to "sparsify" the graph by eliminating edges and then compute an exact LU factorization of the sparsified matrix. This approach provides control over the preconditioner's effectiveness and robustness.

Current techniques to construct support graphs are limited to the class of diagonally dominant M-matrices only. This paper presents a technique to extend these preconditioners to symmetric positive definite matrices obtained from the finite element method. Our approach uses a coordinate transformation at the element level to approximate the coefficient matrix by a symmetric diagonally dominant M-matrix. The quality of this approximation depends only on the mesh topology and not on the problem size. A support graph preconditioner is constructed for this approximation, and used as a preconditioner for the original coefficient matrix. Our approach results in robust preconditioners whose quality is not affected significantly by domain characteristics such as anisotropy and inhomogeneity.

The paper outlines a scheme to construct efficient parallel preconditioners from support graphs. The domain is partitioned into smaller subdomains. Support graphs for the subdomains are augmented with interface edges to form the support graph for the global mesh. The resulting preconditioner can be factorized efficiently on a multiprocessor. The preconditioning step involving triangular system solves can also be parallelized. Our method allows trade-off between the preconditioner's parallelism and its effectiveness, which in turn determines the rate of convergence of the iterative solver. Numerical results show that our preconditioner achieves good speedup on shared-memory multiprocessors such as the IBM p690 and distributed memory multiprocessors such as Beowulf workstation clusters.