Yousef Saad

PHIDAL: A Parallel ILU factorization based on a Hierarchical Interface Decomposition

University of Minnesota
Dept. of Computer Science and Engineering
200 Union st. SE
Minneapolis
MN 55455
saad@cs.umn.edu
Pascal Henon

Ideas from domain decomposition have often been adapted and extended to general sparse matrices to derive parallel solution algorithms for sparse linear systems. The Parallel Hierarchical Interface Decomposition ALgorithm presented in this talk is in this category. The method is reminescent of the the 'wirebasket' techniques of domain decomposition methods [4] and can also be viewed as a variation and an extension of the pARMS algorithm [3] in which the independent sets and levels are defined from a hierarchical decomposition of the interface structure of the graph. From an implementation viewpoint, PHIDAL is an ILU factorization based on a nested dissection-type ordering, in which cross points in the separators play a special role.

The algorithm is based on defining a 'hierarchical interface structure'. The hierarchy consists of classes with the property that Class k nodes, with $k > 0$, are separators for class $k-1$ nodes. In each class, nodes are grouped in independent sets. Class 0 nodes are simply interior nodes of a domain in the graph partitioning of the problem. These are naturally grouped in group-independent sets, in which the blocks (groups) are the interior points of each domain. Nodes that are adjacent to more subdomains will be part of the higher level classes and are ordered last. The factorization uses dropping strategies which attempt to preserve the independent set structure.

One the hierarchical interface decomposition is defined, the Gaussian elimination process proceeds by levels: nodes of the first level are eliminated first, followed by those of the second level etc. All nodes of the first level can be eliminated independently - since there is no fill-in between nodes i and j of two different connectors of level 1. On the other hand fill-ins may appear between connectors at higher levels.

Two options are considered for handling fill-ins. The first is not to allow any fill-in between two uncoupled connectors. Elimination in this case always proceeds in parallel. The second option, which is less restrictive, is to allow fill-ins only between nodes i, j that are in the same processor. In this case, elimination should be done in a certain order and parallel execution can be maintained by exploiting indedepent sets.

Bibliography

[1] R. E. Bank and C. Wagner. Multilevel ILU decomposition. *Numerische Mathematik*, 82(4):543–576, 1999.

[2] E.F.F. Botta and F.W. Wubs. Matrix Renumbering ILU: an effective algebraic multilevel ILU. *SIAM Journal on Matrix Analysis and Applications*, 20:1007–1026, 1999.

[3] Z. Li, Y. Saad, and M. Sosonkina. pARMS: a parallel version of the algebraic recursive multilevel solver. *Numerical Linear Algebra with Applications*, 10:485–509, 2003.

[4] B. Smith, P. Bjørstad, and W. Gropp. *Domain decomposition: Parallel multilevel methods for elliptic partial differential equations*. Cambridge University Press, New-York, NY, 1996.