An Efficient, Numerically Stable, and Scalable Parallel Tridiagonal Solver

Christian Ketelsen, University of Colorado; Markus Berndt, J. David Moulton, T-7

Large tridiagonal systems of linear equations appear in many numerical analysis applications. In our work, they arise in line relaxations needed by robust multigrid methods, such as the parallel BoxMG code [1], for structured grid problems. We present a new numerically stable and scalable algorithm for the solution of diagonally dominant tridiagonal linear systems of equations that scales well on distributed memory parallel computers. Its multilevel design makes it well suited for distributed memory parallel computers with very large numbers of processors.

Background

On a serial computer, a diagonally dominant tridiagonal system of linear equations can be solved in "order N" steps, by using Gaussian elimination and taking advantage of the special sparsity pattern of the linear system. This resulting algorithm is commonly referred to as the Thomas algorithm.

The first parallel algorithm for the solution of tridiagonal systems was developed by Hockney and Golub. It is now usually referred to as cyclic reduction. Stone introduced his recursive doubling algorithm in 1973. Both cyclic reduction and recursive doubling are designed for fine-grained parallelism, where each processor owns exactly one row of the tridiagonal matrix. In 1981, Wang proposed a partitioning algorithm aimed at more coarse-grained parallel computation typical of shared memory clusters, where the number of processors is much smaller than the number of unknowns. There has also been attention directed toward a parallel partitioning of the standard LU decomposition algorithm. In 1986, Sun et al. introduced the parallel partitioning LU algorithm that is very similar to Bondeli's divide and conquer algorithm. These algorithms, while well suited for problems distributed across a moderately large number of processors, do not scale well to very large numbers of processors.

Algorithm

The tridiagonal linear system is assumed to be distributed across a large number of processors, such that each processor owns a contiguous number of rows (see the top matrix schematic of the figure). Each processor generates from that two interface equations that link neighboring processors (indicated by the red rows in the bottom matrix schematic of the figure). This is done efficiently by applying a Cholesky decomposition to the part of the tridiagonal system that is interior to each processor (indicated by the blue matrix entries of the bottom matrix schematic in the figure). The decomposition can be used to generate the upper, as well as the lower interface equation. The new system of interface equations has properties that are similar to the original tridiagonal system. These interface equations are now assigned in large enough groups to a subset of processors, and the resulting tridiagonal system of linear equations is now solved recursively by computing new lower lever interface equations. We proceed with further recursion if the subset of processors is sufficiently large, or solve the new interface system directly on one of the processors in the subset. Once this lowest level set of interface equations is solved, its solution is communicated to the processors

that own the next level up, which then can solve their higher level interface equations, and then on to the highest level, which is the original tridiagonal system of equations. Interestingly, the Cholesky decompositions that were computed along the way to generate each of the interface equations can be re-used now to solve them.

The strength of our new algorithm is that standard numerical components, such as Cholesky decompositions, are used for solving reduced systems of linear equations on each processor. This ensures that the resulting algorithm is numerically stable and easily maintainable. The resulting algorithm can be shown to be numerically stable and to scale well in parallel across very large numbers of processors.

For more information contact Markus Berndt at berndt@lanl.gov.

[1] T.M. Austin, et al., "Parallel, Scalable, and Robust Multigrid on Structured Grids," Los Alamos National Laboratory report LA-UR-03-9167 (2003).

Funding Acknowledgements
NNSA's Advanced Simulation and Computing (ASC), Advanced Applications Program.

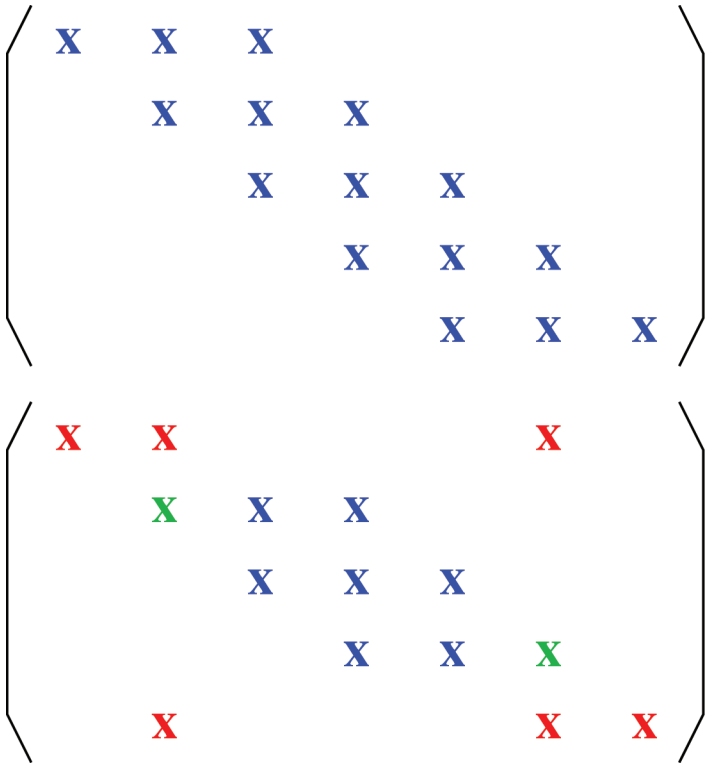

Fig. 1.
Schematic of a tridiagonal matrix owned by one processor before (top) and after the generation of interface equations (bottom).