The Roles of Sparse Direct Methods in Large-scale Simulations

X. S. Li, W. Gao, P. J. R. Husbands, C. Yang, E. G. Ng

One Cyclotron Road, Lawrence Berkeley National Laboratory, MS 50F-1650, Berkeley, CA 94720.

E-mail: {xsli,wggao,pjrhusbands,cyang,egng}@lbl.gov

Abstract. Sparse systems of linear equations and eigen-equations arise at the heart of many large-scale, vital simulations in DOE. Examples include the Accelerator Science and Technology SciDAC (Omega3P code, electromagnetic problem), the Center for Extended Magnetohydrodynamic Modeling SciDAC (NIMROD and M3D-C^1 codes, fusion plasma simulation). The Terascale Optimal PDE Simulations (TOPS) is providing high-performance sparse direct solvers, which have had significant impacts on these applications. Over the past several years, we have been working closely with the other SciDAC teams to solve their large, sparse matrix problems arising from discretization of the partial differential equations. Most of these systems are very ill-conditioned, resulting in extremely poor convergence (sometimes no convergence) for many iterative solvers. we have successfully deployed our direct methods techniques in these applications, which achieved significant scientific results as well as performance gains. These successes were made possible through the SciDAC model of computer scientists and application scientists working together to take full advantage of terascale computing systems and new algorithms research.

1. Overview of SuperLU

SuperLU is a leading scalable solver for sparse linear systems using direct methods, which is partly funded through the TOPS SciDAC project (led by David Keyes) [11]. It is especially targeted for nonsymmetric, indefinite problems. The library routines perform an LU factorization with numerical pivoting and triangular solutions through forward and back substitutions. The factorization can be applied to non-square matrices. In addition, the driver routines contain the functionalities of equilibrating the matrix, reordering the rows and columns of the matrix for stability and sparsity, iterative refinement, estimating the condition number, and computing the forward and backward error bounds. The solver supports both real and complex data types.

The collection of routines has been carefully designed for optimal performance on modern architectures. The sequential algorithm enhances data reuse in memory hierarchy by calling Level 3 BLAS on *supernodes*, which has achieved up to 40% of the peak megaflop rate on many hierarchical memory machines. The shared-memory algorithm exploits both coarse and fine grain parallelism, and employs low-overhead *dynamic scheduling* to minimize parallel runtime. The distributed-memory algorithm enhances scalability using static pivoting and 2D matrix distribution, which lead up to a hundred-fold speedup on large matrices. Moreover, the solver has an easy-to-use and flexible interface to accommodate various needs of the application programs, such as one-factor-multiple-solves and multiple-factors of a sequence of similar matrices, which

occur in the multiple time-step computations. The solver can easily work with any other reordering algorithms not implemented in SuperLU. All these different usage scenarios are realized in one driver routine but with different settings of the input *option* argument. For more details about the algorithms and software architecture, see [7]. In addition to being used in many research codes from the labs and academics, SuperLU is also adopted in commercial use, including Mathematica, FEMLAB, Python, and the HP Mathematical Library, among others.

2. Deployment of SuperLU in fusion simulation codes

The SciDAC Center for Extended Magnetohydrodynamic Modeling (CEMM) [1] is providing the fusion community with flexible and sophisticated tools which can lead to improved understanding of the complex phenomena in fusion reactor plasmas (e.g., in the future ITER experiment), and ultimately lead to better approaches to harnessing fusion energy.

2.1. NIMROD code

NIMROD [9] models fluid-based nonlinear macroscopic electromagnetic dynamics in fusion plasmas. The long-wavelength, low-frequency, nonlinear phenomena involve large-scale changes in the shape and motion of the plasma and severely constrain the operation of fusion experiments. Therefore, the high-fidelity computer modeling is of great desire. The team's primary high-end computing resource is the 10 teraflop/s IBM SP (Seaborg) at the National Energy Research Scientific Computing (NERSC) Center at Lawrence Berkeley National Laboratory.

NIMROD's algorithm uses a high-order finite element representation of the poloidal plane (2D) and a finite Fourier series representation of the toroidal direction (third dimension). The time advance is semi-implicit, which updates the four physical fields separately during each time-step. In three of these advances, different Fourier components (for the periodic coordinate) are not coupled, and a set of systems arising from finite element discretization of the remaining two coordinates (2D) are solved. These correspond to linearized resistive MHD and are relatively small. In the last of the four advances (temperature), the Fourier components are coupled, so there is one larger linear system. The entire calculation is 3D nonlinear resistive MHD, and requires solution of several large sparse linear systems in parallel at every time step. All the matrices involved are complex.

The stiffness inherent in the physical system leads to matrices that are ill-conditioned, since rapid wave-like responses provide global communication within a single time step. The preconditioned conjugate gradient (CG) solver that was used was the most computationally demanding part of the algorithm — took more than 90% of the simulation time. Through SciDAC interactions, we introduced SuperLU to the team, and have been working closely with them for code integration and performance tuning. The performance improvements were dramatic. For 2D linear calculations of MHD instabilities, NIMROD runs more than 100 times faster with SuperLU than it does with the PCG solver. The linear calculations are extremely useful for preliminary explorations that help determine which cases require in-depth nonlinear simulations. For the cutting-edge 3D, nonlinear tokamak simulations (Figure 1), they use a matrix-free CG solver, but compute the elements of an approximate matrix, which is diagonal with respect to Fourier index. The independent submatrices of this approximate matrix are passed to SuperLU, and the approximated systems are solved as the preconditioning step for the matrix-free CG solve. In this case, NIMROD with SuperLU runs 4-5 times faster than before. The nonlinear simulations accumulate relatively small changes in the matrix elements at each time step, but there are no changes to the sparsity pattern. The NIMROD implementation allows SuperLU to reuse its factors repeatedly until the matrix elements accumulate a significant change and refactoring becomes necessary. Refactoring makes the performance improvement less dramatic in nonlinear simulations than in linear calculations, but it is still very significant.

The net result is that computationally demanding simulations of macroscopic MHD

instabilities in tokamak plasmas, which until now were considered too difficult, have become routine.

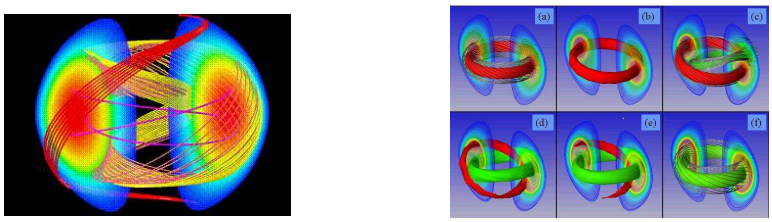


Figure 1. Full 3D numerical simulation of plasma particle drift orbit in a tokamak.

2.2. M3D-C^1 code

M3D is a Multilevel, 3D, parallel, plasma simulation code developed by a multi-institution collaboration [1]. It is suitable for performing linear and nonlinear calculations of plasmas in toroidal topologies including tokamaks and stellarators. M3D-C^1 is a new code being developed by Steve Jardin et al. [4] at Princeton Plasma Physics Laboratory. It performs implicit solution of the two-fluid MHD equations using high-order finite elements with C^1 continuity. The extreme stiffness and anisotropy lead to very ill-conditioned linear systems that posed great challenges to many iterative solvers, including algebraic multigrid method. This naturally pointed to our direct solver. Although the fusion group at PPPL performs production runs on the IBM supercomputer at NERSC, they also acquired a 32-processor SGI Altix system locally to for new code development in order to shorten the turn-around time. For this new 64-bit architecture, we have resolved a word-type inconsistency problem in the interface routines that communicate between SuperLU and M3D-C^1 codes. As a result, the PPPL team can run the largest jobs in a fully parallel mode, with even better than "ideal" scaling. This greatly increased the productivity of the M3D-C^1 development activities, and removed the linear systems solving as a bottleneck.

In the entire nonlinear calculation, the same linear system needs to be solved several times before it changes the nonzero values. The SuperLU interface easily facilitate this need by reusing its factors repeatedly with different right-hand sides.

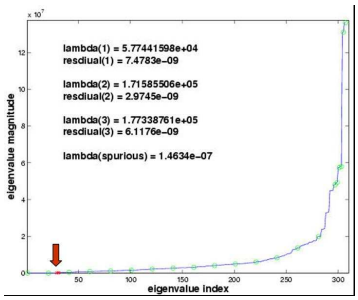
3. Deployment of SuperLU in accelerating cavity modeling

In shape optimization and sensitivity study of next-generation particle accelerators, there is a need to compute approximate cavity resonance frequencies (RF) and the electromagnetic field associated with a cavity structure. Researchers at Stanford Linear Accelerator Center (SLAC) have developed a widely used parallel finite-element code, Omega3P, for electromagnetic modeling of large, complex 3D structures in the frequency domain [5]. The continuum model is represented by the curl-curl formulation of Maxwell's equation in a 3D cavity geometry. The finite-element discretization leads to a generalized eigenvalue problem $Kx = \lambda Mx$, where K is real symmetric and M is symmetric and positive definite. The problem is challenging because of the wide distribution of the spectrum, the need for finding the relatively small interior eigenvalues, and the large problem size (a high resolution modeling requires millions of mesh points). Typically, there are large number of eigenvalues very close to zero. Only the first few (tens to hundreds) smallest "nonzero" eigenvalues and their associated eigenvectors are sought. Figure 2 (a) shows the eigenvalue distribution of a small eigensystem.

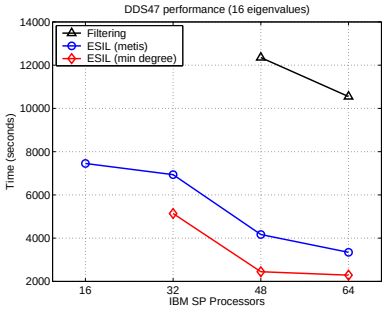
An effective method to find these small interior eigenvalues is by spectral transformation, which yields the new eigensystem: $M(K - \sigma M)^{-1}Mx = \mu Mx$, with $\mu = \frac{1}{\lambda - \sigma}$. The largest eigenvalues of the new system correspond to the original eigenvalues closest to the shift σ, and

they are well separated from the others. At each iteration of the Lanczos algorithm, a shifted linear system involving operator $(K-\sigma M)^{-1}$ must be solved. The earlier algorithm in Omega3P is called the Filtering algorithm [10], which solves the linear system iteratively, and in the end uses a Newton-type correction to refine the eigen-approximation. We implemented an efficient Exact Shift and Invert Lanczos (ESIL) solver by coupling parallel SuperLU [8] with parallel ARPACK [6], in which SuperLU is used for direct solution of the linear system $(K-\sigma M)x = b$. This enabled accurate calculation of eigenvalues and allows verification of the other eigensolvers. Figure 2 (b) compares the parallel runtimes of the two algorithms for a 47-cell DDS cavity structure, for which the matrix dimension is 1.3×10^6 and 16 eigenpairs close to a specified shift were computed. For ESIL, we used two different reordering methods: METIS and minimum degree. It is clear that although each iteration of ESIL is more expensive than Filtering, many fewer iterations are needed by ESIL (about 9 iterations per eigenvalue), and it is 2 to 3 times faster than Filtering.

The largest system solved is of order 7.5 million, with 304 million nonzeros in each matrix. Using one shift, we were able to find 10 eigenvalues close to that shift. PARPACK needs $\tilde{5}.5$ solve for each eigenvalue. The entire eigensolver time is about 2.5 hours using 24 processors of Seaborg.



(a) Eigenvalue distribution



(b) Parallel runtimes

Figure 2. (a) Eigenvalue distribution of an accelerator model problem. The eigenvalues sought are in the region marked by the arrow. (b) Parallel runtimes of Filtering and ESIL algorithms on the IBM SP at NERSC.

4. Recent and future developments driven by SciDAC applications

Most of our recent algorithmic research and software developments were motivated by the needs from SciDAC applications. In particular, the large scale of the problem revealed some bottlenecks in the solver that were not detected while testing smaller problems, and our SciDAC-funded work has been mostly in parallel algorithms/codes.

For parallel SuperLU, we re-designed the matrix redistribution algorithm that transforms the user input format into the internal distributed format, which led to 10-20 times faster for the largest problems in accelerator modeling. We designed a parallel symbolic factorization algorithm which works directly on the distributed input matrix without the need to gather the distributed graph onto one single processor. The parallelism achieved 5 times reduction in per-processor memory usage and 12 times speedup [3]. This removed the memory bottleneck for the largest problems. We designed a new reordering algorithm for nonsymmetric matrices which respects the unsymmetric structure and preserves sparsity better — up to 22% reduction in factor size was observed.

When modeling the full size accelerator structure, the eigenvalue problems can reach dimension of millions. The ESIL method would be prohibitively expensive because of the need for entire factorizations. This prompted us to look into the Algrbraic Multilevel Sub-structuring Algorithm (AMLS). AMLS is very attractive for largest eigenvalue problems because

of its reminiscence of domain decomposition technique successfully used for linear systems. An important step is the congruence transformation resulting in many decoupled subproblems. This step requires block Gaussian elimination but at a coarse-grain level, and is therefore amenable to a scalable implementation. The techniques in sparse direct methods are very useful here. We developed an efficient algorthm and code that can handle large null space specific in the accelerator problems. We analyzed the approximate properties of the algorithm, investigated the complexity of its implementation, and devised better mode selection criteria to improve its efficiency, and demonstrated its performance advantage over ESIL for the accelerator problems [12].

In addition to improving SuperLU, we are exploring some asymtotically faster algorithms. We found that many SciDAC applications involve elliptic PDEs, for which the inverse of the continuous operator looks like an integral operator with the kernel smooth away from the diagonal, so the off-diagonal blocks of the discretized inverse operator have low numerical-ranks. We can rapidly compute the low-rank structures on the fly using a compressed representation such as SVD, which also exhibits high locality. The resulting factorization algorithm would have *almost-linear* time complexity, breaking the theoretical lower bound of classical Gaussian elimination algorithm [2]. The innovative ideas to linear system are analogous to fast multipole methods to matrix-vector multiplication, and will have great impact for tera-scale PDE computations.

5. Acknowledgments

We wish to thank many members of the SciDAC collaborating teams, in particular, Steve Jardin and Joshua Breslau from Princeton Plasma Physics Laboratory, Carl Sovinec from University of Wisconsin-Madison, and Kwok Ko and Rich Lee from Stanford Linear Accelerator Center. Their enthusiasm and the engaged in-depth discussions of the physics problems greatly helped us improve the algorithm design.

6. References

[1] Center for Extended MHD Modeling (CEMM). URL: http://w3.pppl.gov/cemm/.

[2] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, and A-J. van der Veen. Fast stable solver for sequentially semi-separable linear systems of equations. *SIAM J. Matrix Analysis and Applications*, 2005. to appear.

[3] Laura Grigori, James W. Demmel, and Xiaoye S. Li. Parallel symbolic factorization algorithm. Technical report, Lawrence Berkeley National Laboratory, 2005. in preparation.

[4] S. C. Jardin and J.A. Breslau. Implicit solution of the 4-field extended-magnetohydrodynamic equations using high-order high continuity finite elements. Technical report, Plasma Physics Laboratory, Princeton University, 2004.

[5] K. Ko, N. Folwell, L. Ge, A. Guetz, V. Ivanov, L. Lee, Z. Li, I. Malik, W. Mi, C. Ng, and M. Wolf. Electromagnetic systems simulation - from simulation to fabrication. SciDAC report, Stanford Linear Accelerator Center, Menlo Park, CA, 2003.

[6] Rich Lehoucq, Kristi Maschhoff, Denny Sorensen, and Chao Yang. Parallel ARPACK. http://www.caam.rice.edu/∼kristyn/parpack_home.html.

[7] Xiaoye S. Li. An overview of SuperLU: Algorithms, implementation, and user interface. *ACM Trans. Mathematical Software*, 2005. to appear.

[8] Xiaoye S. Li and James W. Demmel. SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems. *ACM Trans. Mathematical Software*, 29(2):110–140, June 2003.

[9] C.R. Sovinec, A.H. Glasser, D.C. Barnes, T.A. Gianakon, R.A. Nebel, S.E. Kruger, D.D. Schnack, S.J. Plimpton, A. Tarditi, M.S. Chu, and the NIMROD Team. Nonlinear magnetohydrodynamics with high-order finite elements. *Computational Physics*, 195:355, 2004.

[10] Yong Sun. *The Filter Algorithm for Solving Large-Scale Eigenproblems From Accelerator Simulations*. PhD thesis, Department of Computer Science, Stanford University, March 2003.

[11] Terascale Optimal PDE Simulation. URL: http://www-unix.mcs.anl.gov/scidac-tops/.

[12] C. Yang, W. Gao, Z. Bai, X. Li, L. Lee, P. Husbands, and E. Ng. An algebraic sub-structuring method for large-scale eigenvalue calculation. *SIAM J. on Scientific Computing*, 2005. to appear.