The Roles of Direct Methods in Large-Scale Simulations

Abstract

TOPS is providing high-performance, scalable sparse direct solvers, which have had significant impacts on the SciDAC applications, including fusion simulation (CEMM), accelerator modeling (AST), as well as many other mission-critical applications in DOE and elsewhere.

Our developments attempt to meet the new challenges in the increasingly difficult problems (3D geometry, higher resolution, multiscale physics, anisotropy, elasticity, . . .)

- The systems can be very ill-conditioned such that many iterative solvers do not work well. The dimensions are large (# DOF's can reach tens to hundreds of millions) in the interest of high-fidelity simulations/modeling.
- Need improvements to the existing algorithms/codes to reduce inter-processor communication and increase scalability. Need develop new algorithms/codes that have lower algorithmic complexity.

SuperLU

(http://crd.lbl.gov/~xiaoye/SuperLU)

SuperLU is a leading scalable sparse direct solver

Rich functionalities

- LU factorization with numerical pivoting, triangular solutions
- Reordering algorithms for stability and sparsity
- Pre-equilibrate the badly-scaled systems
- Iterative refinement
- Condition number estimation, forward and backward error bound

High-performance architecture awareness

- Sequential SuperLU – enhances data reuse in memory hierarchy by calling Level 3 BLAS on the supernodes
- SuperLU_MT (shared-memory) – exploits both coarse and fine grain parallelism; employs dynamic scheduling to minimize parallel runtime
- SuperLU_DIST (MPI-based) – enhances scalability by static pivoting and 2D matrix distribution

Adopted by FEMLAB, HP Math Library, Mathematica, Python, NAG

In many DOE projects: fusion (M3D, NIMROD), accelerator (Omega3P), supernova (PHOENIX), Trilinos, PETSc, NIKE, . . .

New Developments Driven by SciDAC Needs

Better matrix redistribution algorithm led to 10-20 × faster for the largest problems in accelerator modeling

Parallelized the symbolic analysis phase

- Removed memory bottleneck for the largest problems
- 5 × reduction in per-processor memory usage and 12 × speedup

Better reordering for unsymmetric matrices

- Constrained Markowitz algorithm with Local Symmetrization
- Respects the unsymmetric structure in the matrix
- 22% reduction in factor size

TOPS and CEMM sped up fusion code NIMROD

NIMROD is a parallel fusion plasma modeling code using fluid-based nonlinear macroscopic electromagnetic dynamics.

Solve the continuum equations of magnetohydrodynamics, using discretizations of higher than second order on unstructured mesh.

Complex, unsymmetric sparse linear systems, hundreds of thousands unknowns, very ill-conditioned. (>90% execution time)

Swapping in parallel SuperLU to replace diagonally scaled Krylov method led to dramatic improvement on NERSC IBM SP.
2D tests run >100 × faster; 3D production runs are 5-10 × faster

"Equivalent of 3-5 years progress in hardware"
– Dalton Schnack, SAIC

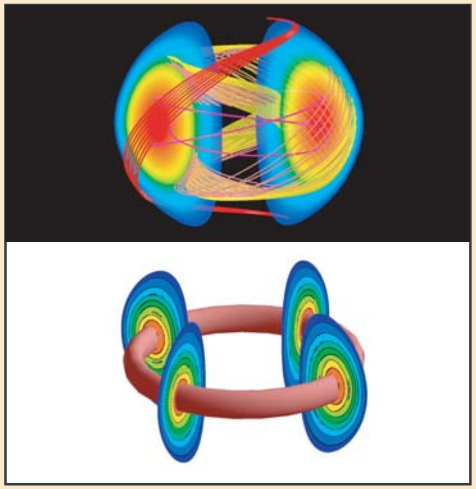


Figure 1. Full 3D numerical simulation of plasma particle drift orbits in a tokamak (top), and a particular configuration (bottom).

TOPS and CEMM Collaborating on M3D-C1 Code

M3D-C1 is a new code being developed at CEMM

- Implicit solution of the two-fluid MHD equations using high-order high-continuity finite elements. (vs. explicit scheme in the old M3D.)
- Extreme stiffness and anisotropy leads to very ill-conditioned linear systems – real nonsymmetric, tens to hundreds of thousands unknowns

Integrate parallel SuperLU into M3D-C1 on the SGI Altix at Princeton Plasma Physics Laboratory

- Able to run largest jobs in a fully parallel mode, with even better than "ideal" scaling
- Increased the code-development productivity for M3D-C[1]

Removed linear systems solutions as a bottleneck!

TOPS and AST Collaborating on Modal Analysis

The design of accelerator cavities requires solutions of Maxwell's equations

$$\begin{aligned} \nabla\times(\nabla\times\mathbf{E}) - \lambda\mathbf{E} &= 0 \ \text{ in } \Omega \\ n\times\mathbf{E} &= 0 \ \text{ on } \Gamma_E \\ n\times(\nabla\times\mathbf{E}) &= 0 \ \text{ on } \Gamma_B \end{aligned} \quad \xRightarrow{\mathbf{E}=\sum_i x_i N_i} \quad K\,x = \lambda\,M\,x$$

FEM in frequency domain leads to large sparse generalized eigensystem

Need to understand the wake field in the full structure

- Want large number of low frequency modes and field vectors, i.e., smallest nonzero eigenvalues and may be tightly clustered

Figure 2. A 47-cell full structure

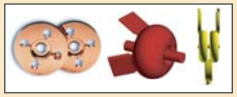


Figure 3. Examples of individual cells

3D structures, high resolution simulations ⟶ extremely large matrices

- Real symmetric for lossless structure, complex symmetric for lossy structure

Need high accuracy – tolerate only 0.01% error

Parallel Shift-Invert Lanczos algorithm

$$M\,(K - \sigma\,M)^{-1}\,x = \mu\,M\,x, \quad \mu = 1\,/\,(\lambda - \sigma)$$

- Each step requires solution of sparse linear systems
- SLAC (Omega3P): Inexact linear solver + Newton correction
- TOPS contribution (ESIL): combine parallel SuperLU (exact linear solver) and parallel ARPACK (Lanczos)

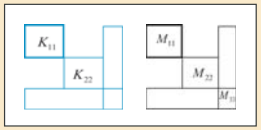
Enable accurate calculation of eigenvalues, allow verification of other eigensolvers

Our ESIL is more reliable, and often 2-3 × faster

Alternative Eigensolver – Multilevel Substructuring

Parallelism abounds due to "domain-decomposition" . . .

1. Partition & reorder (K,M)

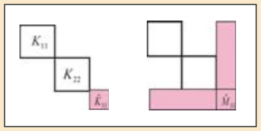


2. Block Gaussian elimination (congruence transformation)

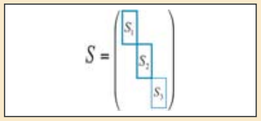
$$\hat{K} = L^{-1}KL^{-T} \qquad \hat{M} = L^{-1}ML^{-T}$$

3. Sub-structure calculation

$$K_{ii}v^{(i)} = \mu^{(i)}M_{ii}v^{(i)}, \quad i = 1,2$$
$$\hat{K}_{33}v^{(3)} = \mu^{(3)}\hat{M}_{33}v^{(3)}$$



4. Subspace assembling



5. Rayleigh-Ritz projection

$$\left(S^T\hat{K}S\right)q = \theta\,(S^T\hat{M}S)\,q$$

Future Plans in Sparse Direct Solvers

Improve SuperLU overall speed, esp. triangular solve

- "Switch to dense" towards the end
- Incorporate Berkeley OSKI automatic tuning techniques

Explore asymptotically faster algorithms . . .

Almost-linear time direct solver and preconditioner

- Key observation: for elliptic PDEs, 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

- Rapidly compute the low-rank structures on the fly in linear time (e.g., using a compressed representation by SVD)
- More dramatic for 3D problems!
- Innovative ideas to linear systems are analogous to fast multipole methods to matrix-vector multiplication