11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
July 20–25, 2014, Barcelona, Spain

FAST MULTIPOLE PRECONDITIONERS FOR SPARSE LINEAR SOLVERS

H. Ibeid[1], R. Yokota[1], J. Pestana[2], and D. Keyes[1]

[1] King Abdullah University of Science and Technology, 4700 KAUST, Thuwal, Saudi Arabia,
huda.ibeid;rio.yokota;david.keyes@kaust.edu.sa
[2] University of Oxford, Andrew Wiles Building, Woodstock Road, Oxford, OX2 6GG UK,
jennifer.pestana@maths.ox.ac.uk

Key words: *Fast Multipole Method; Preconditioner; Krylov Subspace Method*

The current trend in computer architecture favors algorithms with data-locality and asynchronicity. Unfortunately, elliptic PDEs are global in nature since some form of information must travel from one end of the domain to the other. Algorithmically, this can be achieved through either successive local communications by using iterative methods, or a direct global communication by using direct solvers. Iterative methods achieve data-locality of each communication but require many synchronizations to assure the propagation of information through these local communications. Conversely, direct methods require global communication, which can be performed asynchronously. In most cases, these two approaches are combined as a preconditioned Krylov solver to achieve both data-locality and asynchronicity.

Improving the data-locality and asynchronicity of Krylov iterations is an interesting topic all on its own [1], but in the present work we focus on the preconditioner. The three important keywords in our approach are *hierarchical*, *approximate*, and *matrix-free*. The *hierarchy* brings the arithmetic complexity down from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ and the communication complexity down from $\mathcal{O}(P)$ to $\mathcal{O}(\log P)$, where N is the number of unknowns and P is the number of processes. The *approximation* decreases the asymptotic constant of $\mathcal{O}(N)$ and $\mathcal{O}(\log P)$ by trading off the accuracy. These features are shared by methods like multigrid [2] and $\mathcal{H}$-matrices [3]. The third feature *matrix-free* will further decrease the asymptotic constant of the communication, but will increase the asymptotic constant of the computation. Such an approach will make more and more sense as the computation becomes cheaper compared to data movement on future architectures. The combination of these three features can be achieved by using the Fast Multipole Method (FMM) [4] as a preconditioner.

For the FMM to be used as a preconditioner, there are a few issues that need to be resolved. First of all, the original free-space boundary condition of the FMM must be

extended to finite domain boundary conditions. In the present work we use the boundary integral equation (BIE) to enable this extension. We would like to emphasize the distinction between the present method and methods that use the FMM for the matrix-vector multiplication within the Krylov solvers for BIEs. The main part of our calculation is the volume-to-volume contribution and the BIE is only a small portion of our calculation. Our method is used to precondition sparse matrices arising from finite difference/element discretizations and not boundary element discretizations, and can handle a much broader range of scientific applications.

When using FMM as a preconditioner the accuracy requirements are lower compared to when it is used as a direct solver. This allows us to perform low accuracy optimizations such as the use of Cartesian expansions, dual tree traversals with error optimized multipole acceptance criteria, single precision variables, and low precision SIMD transcendental functions. Comparisons with other FMM codes have revealed that these low accuracy optimizations can result in orders of magnitude of increase in performance [5]. Our FMM code also makes use of all forms of parallelism that is available by combining MPI, pthreads, OpenMP, Intel TBB, SSE, AVX, and MIC intrinsics, depending on the architecture.

We applied our FMM preconditioner to a Poisson problem and Stokes problem using finite elements by plugging it into PETSc via PetIGA [6]. Using this framework, we are able to switch between our FMM preconditioner and an algebraic multigrid preconditioner – BoomerAMG– with the runtime option "`--pc_type`" in PETSc. The FMM preconditioner shows similar convergence rate to the multigrid preconditioner when the FMM accuracy is set to $\epsilon = 10^{-4}$, which corresponds to four significant digits of accuracy [7]. The FMM preconditioner shows better scalability and becomes faster than multigrid past 16 cores [7].

REFERENCES

[1] M. Hoemmen. Communication-Avoiding Krylov Subspace Methods. *PhD Thesis, U. C. Berkeley*, 2010.

[2] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, London, 2001.

[3] W. Hackbusch. A Sparse Matrix Arithmetic Based on $\mathcal{H}$-matrices, Part I: Introduction to $\mathcal{H}$-matrices. *Computing*, 62:89–108, 1999.

[4] L. Greengard and V. Rokhlin. A Fast Algorithm for Particle Simulations. *Journal of Computational Physics*, 73:325–348, 1987.

[5] R. Yokota. An FMM Based on Dual Tree Traversal for Many-core Architectures. *Journal of Algorithms and Computational Technology*, 7:301–324, 2013.

[6] V. Calo, N. Collier, and L. Dalcin. PetIGA: High-performance isogeometric analysis. *arxiv:1305.4452*, 2013. http://arxiv.org/abs/1305.4452.

[7] R. Yokota, J. Pestana, H. Ibeid, D. Keyes. Fast Multipole Preconditioners for Sparse Matrices Arising from Elliptic Equations. *arxiv:1308.3339*, 2013. http://arxiv.org/abs/1308.3339.