On the GPU/CPU Implementation of Direct Domain Decomposition Methods (D^3M)

J. Moshfegh, D. G. Makris, and M. N. Vouvakis
Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA, USA
Email: {moshfegh, vouvakis}@ecs.umass.edu

General-purpose computing on graphics processing units (GPUs) has gain significant traction in High Performance Computing (HPC) and Artificial Intelligence (AI) communities. In HPC and computational science and engineering communities GPUs and GPU/CPU have been heavily used in dense matrix computations, due to the fact that a dense matrix is a form of an image, one with real or complex-valued pixels. One fine example of dense linear algebra and GPU symbiosis is the MAGMA dense linear algebra library, which implements in GPU almost the entire LAPACK library functionality (S. Tomov and J. Dongarra and M. Baboulin, Towards dense linear algebra for hybrid GPU accelerated manycore systems, Parallel Computing, no. 5-6, pp. 232-240, Jun 2010). Contrary, sparse matrix linear algebra and its applications are much harder to tailor to GPU architectures due to the irregular, non-local pattern of the input data, that gets exacerbated by fill-ins and pivoting. To bypass this difficulty researchers in finite element electromagnetic computations have opted for FEM solvers that rely on iterative matrix solver GPU implementations with somewhat rudimentary direct solver preconditioners (H. Meng, B. Nie, S. Wong, C. Macon and J. Jin, "GPU accelerated finite-element computation for electromagnetic analysis," in IEEE Antennas and Propagation Magazine, vol. 56, no. 2, pp. 39-62, April 2014). Nonetheless, sparse direct solvers such as Cholesky factorization have been attempted on GPUs and indeed show sizable gains over CPU implementations, but certainly much more modest than those on dense matrix counterparts (S. Rennich, D. Stosic, and T. Davis, Accelerating sparse Cholesky factorization on GPUs, 4th Workshop on Irregular Applications: Architectures and Algorithms (IA3 '14)).

This paper aims to bridge the direct sparse preference to CPUs with that of the direct dense matrix solver preference GPUs, with using the memory efficient direct DDM (D^3M) (J. Moshfegh, and M. N. Vouvakis, IEEE APS, 2017). Unlike other sparse direct matrix solvers, D^3M uses many "embarrassingly parallel" small sparse matrix factorizations to intentionally form another small but also block-wise sparse matrix that must be factorized with a special block LDL^T direct method. This LDL^T factorization has the dominant contribution in D^3Ms run-time. Since this matrix is a block-wise sparse, with relatively large block dimensions ($n > 300$), it is quite suitable for GPU computations. We aim to use cuBLAS (CUDA basic linear algebra subroutines library from NVIDIA Corporation) and GPUs to improve the performance of this factorization step. Using this hybrid CPU and GPU implementation, we hope to achieve significant time savings. Results illustrating the validity and efficiency of the method on real-world scattering and radiation problems will be presented.