Using low-rank approximation techniques for engineering problems

J. Anton[1], C. Ashcraft[1], Pierre L'Eplattenier[1], R. Grimes[1], F.-H. Rouet[1,a)] and C. Weisbecker[1]

[1]Livermore Software Technology Corporation, 7374 Las Positas Road, Livermore, CA 94551, USA

[a)]Corresponding and presenting author: fhrouet@lstc.com

ABSTRACT

Matrices that appear in the boundary element methods and finite element methods are often *structured* (or *low-rank*, or *data-sparse*) [3, 5]. This means that they exhibit rank-deficient blocks, typically the blocks corresponding to far range interactions in the physical space. Identifying and compressing these low-rank blocks, e.g., using SVD or a rank-revealing factorization, is the key to reducing the storage and computational requirements of many matrix operations, such as performing matrix-vector products, computing eigenvalues, and solving linear systems. In this talk, we will focus on the latter, for both dense and sparse matrices. For sparse matrices, the low-rank property is usually not found in the input matrix but at intermediate steps of the factorization algorithms used to solve linear systems.

Many different techniques, referred to as *low-rank representations*, have been proposed in the literature. Among others, the Hierarchically Semi-Separable (HSS) matrices [8] and Block Low-Rank representations [2] have been widely studied and have recently been implemented in parallel solvers. However very few comparison results can be found in the literature; usually they are restricted to model problems, or to comparing a single low-rank algorithm against a non-low-rank one. Our goal is to compare the performance of the HSS and BLR approaches for dense and sparse matrices arising from engineering applications.

LS-DYNA [6] is a highly advanced nonlinear finite element code. It allows implicit and explicit simulations of multi-physics problems, such as mechanics, fluid dynamics, acoustics, electromagnetism... It is widely used by the automotive, aerospace, and construction industries among others. The matrices that we will consider for this presentation all arise from implicit simulations performed with LS-DYNA for real world applications. We will compare the HSS and BLR techniques using multiple high-performance implementations. The HSS-based solver we will use is the STRUMPACK code [4] [7], that can be used as a preconditioner or as a direct solver for both dense and sparse problems. For BLR we will use MUMPS [1] [2], a sparse direct solver that has recently gained Block Low-Rank features.

Keywords: Linear systems, dense matrices, sparse matrices, low-rank approximations, finite elements.

References

[1] Amestoy, P. R., Guermouche, A., L'Excellent, J.-Y. and Pralet, S. (2006) Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, **32 (2)**, 136-156. **50**, 937-951.

[2] Amestoy, P. R., Ashcraft, C., Boiteau, O., Buttari, A., L'Excellent, J.-Y. and Weisbecker, C. (2015) Improving multifrontal methods by means of block low-rank representations. *SIAM Journal on Scientific Computing*, **37 (3)**, A1451-A1474.

[3] Chandrasekaran, S., Dewilde, P., Gu, M. and Somasunderam, N. (2010) On the numerical rank of the off-diagonal blocks of Schur complements of discretized elliptic PDEs. *SIAM Journal on Matrix Analysis and Applications*, **31 (5)**, 2261–2290.

[4] Ghysels, P., and Li, X. S., Rouet, F.-H., William, S. and Napov, A. (2014). An efficient multi-core implementation of a novel HSS-structured multifrontal solver using randomized sampling. *SIAM Journal on Scientific Computing*, submitted.

[5] Hackbusch, W. and Khoromskij, B. N. (2000) A sparse $\mathcal{H}-$matrix arithmetic. *Computing*, **64 (1)**, 21–47.

[6] Hallquist, J. O., et al. (1992) LS-DYNA theory manual. *Livermore Software Technology Corporation*

[7] Rouet, F.-H., Li, X. S., Ghysels, P. and Napov, A. (2015) A distributed-memory package for dense hierarchically semi-separable matrix computations using randomization. *ACM Transactions on Mathematical Software*, submitted.

[8] Xia, J., Chandrasekaran, S., Gu, M. and Li, X. S. (2010) Fast algorithms for hierarchically semiseparable matrices. *Numerical Linear Algebra with Applications*, **17 (6)**, 953–976.