ParCFD'2020
32^{nd} International Conference on Parallel Computational Fluid Dynamics
May-11-13 2020, Nice FRANCE

ACCELERATING 3D UNSTRUCTURED MESH DEFORMATION BASED ON RADIAL BASIS FUNCTIONS INTERPOLATIONS USING TILE LOW-RANK MATRIX COMPUTATIONS

Hatem LTAIEF[*], Youssef MESRI[†], Rabab ALOMAIRY[*], Wael BADER[†], and David KEYES[*]

[*] Extreme Computing Research Center, KAUST, 23955 Thuwal, Saudi Arabia
e-mail: first.author@kaust.edu.sa, web page: https://cemse.kaust.edu.sa/ecrc

[†] MINES ParisTech, PSL University, 06604 Sophia Antipolis, France
e-mail: first.author@mines-paristech.fr, web page: https://www.hpc-ai.mines-paristech.fr

Key words: Fluid-Structure Interaction, 3D Mesh Deformation, Radial Basis Functions, Low-Rank Matrix Computations.

Abstract. We leverage the performance of 3D unstructured mesh deformation in the context of fluid-structure interactions. We employ the Radial Basis Function (RBF) interpolations as a well-known numerically robust approach that produces deformed meshes with high fidelity. The resulting operator is a dense symmetric matrix of size N, with N the number of nodes in the boundary of the mesh. The cubic arithmetic complexity and the quadratic memory footprint often make the system challenging to solve using a direct method. In this paper, we accelerate the computations of 3D unstructured mesh deformation based on RBF interpolations using tile low-rank matrix computations. The idea consists in exploiting the data sparsity of the matrix operator of the linear system by approximating off-diagonal tiles up to an application-dependent accuracy threshold. We demonstrate the effectiveness of our implementation by assessing the numerical accuracy of the mesh deformation. We then provide preliminary performance benchmarking on two shared-memory systems. The high performance tile low-rank solver permits to achieve up to 20-fold performance speedup over the state-of-the-art dense matrix solvers.

1 INTRODUCTION

The simulation of physical phenomena involving moving bodies undergoing large mesh deformations represents a complex challenge. Indeed, the deformed mesh often has poorer quality than the initial one, which may lead to inconsistent and unstable numerical solutions. The mesh quality may get even worse during the time integration since its elements may become inverted or highly skewed, which may result in invalidating the mesh needed to perform simulation. Mesh deformation is a standard method for solving the aforementioned issue. It aims at deforming the mesh in order to track the moving geometries [6]. It uses using r-adaptation techniques (i.e., modifying only the mesh point coordinates and leaving the connectivity unchanged) driven by either physical or numerical analogies. This approach is often coupled to mesh optimization operations to maintain a good quality of the mesh, while deforming it using either local re-meshing operations such as connectivity change (vertex insertion/deletion and face/edge swapping) and smoothing (vertex displacements) [3]. In this paper, we study the Radial Basis Function (RBF) [7] scheme, one of the main interpolation methods that produces meshes with high fidelity. The RBF technique is computationally expensive for tracking 3D moving objects since it requires solving at each time step a large dense linear system. In particular, we accelerate the solver by exploiting the data sparsity of the symmetric positive-definite matrix operator to reduce the memory footprint as well as the arithmetic complexity. Our preliminary results demonstrate a performance gain of our Tile Low-Rank (TLR) Cholesky-based solver against the state-of-the-art dense solvers from optimized libraries on two shared-memory systems.

2 RADIAL BASIS FUNCTION INTERPOLATIONS FOR 3D UNSTRUCTURED MESH DEFORMATION

Radial basis function (RBF) interpolation is used here to describe the displacement of the internal volume nodes given the displacement of the boundary nodes. As described in [4], the displacement $\tilde{d}$ in the whole domain, can be approximated by a sum of basis functions as follows: $\tilde{d}(x) = \sum_{i=1,n_b} \alpha_i \phi(||x - x_{b_i}||) + p(x)$, where $x_{b_i} = [x_{b_i}, y_{b_i}, z_{b_i}]$ are the boundary nodes in which the values are known, p a polynomial, n_b the number of boundary nodes and ϕ a given basis function. The coefficients α_i and the polynomial p are determined by the interpolation conditions $\tilde{d}(x_{b_i}) = d_{b_i}$, where d_b contains the known displacement values at the boundary. There are additional requirements $\sum_{i=1,n_b} \alpha_i q(x_{b_i}) = 0$ for all polynomials q with a $deg(q) \leq deg(p)$. The minimal degree of polynomial p depends on the choice of the basis function ϕ. A unique interpolant is given if the basis function is a conditionally positive definite function. If the basis functions are conditionally positive definite of order $m \leq 2$, a linear polynomial can be used. Herein, we only consider basis functions that satisfy this criterion. Although the size of the linear system is much smaller than the number of degrees of freedom for the overall mesh deformation problem, the arithmetic complexity remains a computational challenge for large 3D moving objects. An iterative solver may fail for matrices with high condition number or may simply render inefficient in presence of multiple right hand sides. Instead, we employ a direct dense solver that takes advantage of the data sparsity of the symmetric positive-definite matrix to reduce the time complexity.

3 TILE LOW-RANK MATRIX COMPUTATIONS

We solve the resulting RBF linear system using the Tile Low-Rank (TLR) Cholesky factorization, as implemented in the `HiCMA` library [1]. The main idea of `HiCMA` consists in approximating the off-diagonal tiles of the symmetric positive-definite matrix up to an application-dependent accuracy threshold. Once the matrix approximated using typically the Randomized Singular Value Decomposition (RSVD), the TLR Cholesky can then

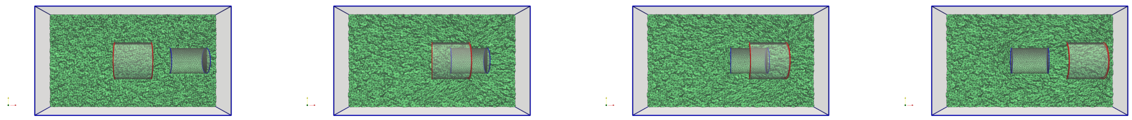
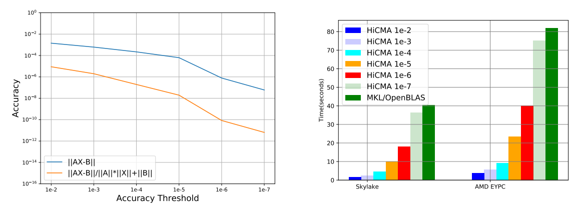


Figure 1: Snapshots of the mesh deformations for two interpenetrating 3D cylinders.

operate on each logical tile, represented in terms of compressed data structures. `HiCMA` relies on the dynamic runtime system `StarPU` to orchestrate and asynchronously schedule the various computational tasks onto the underlying hardware resources.

4 PRELIMINARY ASSESSMENTS



(a) Accuracy results. (b) Performance results.

Figure 2: Numerical and performance assessments.

The double precision carried experiments have been done on two shared-memory systems: a two-socket 20-core Intel Xeon Gold Skylake 6148 CPUs @ 2.40 GHz and an AMD Epyc system with two-socket 32-core 7601 CPUs @ 2.2GHz. For BLAS and LAPACK implementations, MKL (v2018) and OpenBLAS (v0.2.20) are used on the Intel and AMD systems, respectively. Figure 1 shows two cylinders interpenetrating each other with only a thin layer of elements in-between. Both cylinders are placed in a rectangular domain of dimensions of 98668 vertices and 536944 tetrahedra. The size of the corresponding RBF's linear system to solve is 55K. Figure 2(a) reports the numerical results using the infinite norm for several accuracy thresholds, i.e, from $1e-2$ to $1e-7$. As expected, the error is of the same order than the accuracy threshold and linearly decreases as the accuracy threshold. We plan to report numerical results for lower accuracy since the application is resilient to further digit losses. However, a TLR LU-based solver may have to be considered since the symmetric matrix may loose its positive-definiteness. Figure 2(b) highlights preliminary results of TLR Cholesky applied to the RBF matrix operator on the Intel and AMD architectures. For the 55K matrix size considered, `HiCMA` TLR Cholesky achieves up to a 20-fold speedup against dense Cholesky from MKL and OpenBLAS implementations on the Intel and AMD systems, respectively. We have employed a matrix scaling phase in order to restore the positive definiteness of the RBF matrix encountered for low accuracy thresholds, as explained in [5]. We plan to run against larger RBF matrices and scale up to distributed-memory systems. Figure 3 depicts the rank heatmaps for the studied accuracy thresholds, after the matrix compression phase and after the Cholesky factorizations. Since the tiles with high ranks are located at the top left corner of the symmetric matrix after compression, the rank growth remains limited after Cholesky computation and only impacts the top left block of tiles.

5 CONCLUSIONS

We have accelerated up to a 20-fold speedup the 3D unstructured mesh deformation based on radial basis functions interpolations using tile low-rank Cholesky computations. We plan to study larger 3D moving objects and compare performance / accuracy against other low-rank matrix approximations [2].

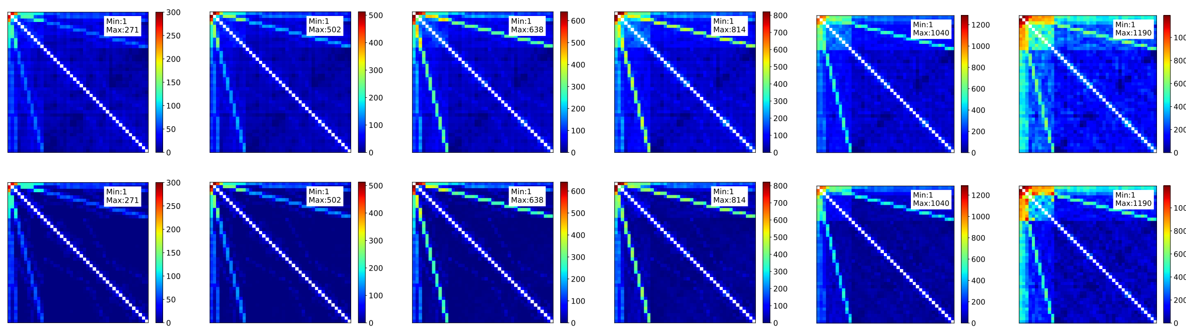


Figure 3: Rank heatmaps of the RBF matrix operator: initial (top row) and final (bottom row) ranks, varying in column from accuracy $1e-2$ to $1e-7$.

REFERENCES

[1] K. Akbudak, H. Ltaief, A. Mikhalev, and D. Keyes. Tile Low Rank Cholesky Factorization for Climate/Weather Modeling Applications on Manycore Architectures. In *International Supercomputing Conference*, pages 22–40. Springer, 2017.

[2] P. Coulier and E. Darve. Efficient Mesh Deformation based on Radial Basis Function Interpolation by Means of the Inverse Fast Multipole Method. *Computer Methods in Applied Mechanics and Engineering*, 308:286–309, 2016.

[3] C. Dapogny, C. Dobrzynski, and P. Frey. Three-dimensional Adaptive Domain Remeshing, Implicit Domain Meshing, and Applications to Free and Moving Boundary Problems. *Journal of Computational Physics*, 262:358 – 378, 2014.

[4] A. de Boer, M. van der Schoot, and H. Bijl. Mesh Deformation Based on Radial Basis Function Interpolation. *Computers and Structures*, 85(11):784 795, 2007. Fourth {MIT} Conference on Computational Fluid and Solid Mechanics.

[5] N. Higham and S. Pranesh. Exploiting Lower Precision Arithmetic in Solving Symmetric Positive Definite Linear Systems and Least Squares Problems. *Technical Report, http://eprints.maths.manchester.ac.uk/id/eprint/2736*, 2019.

[6] E. Lefrançois. A Simple Mesh Deformation Technique for Fluid-structure Interaction based on a Submesh Approach. *International Journal for Numerical Methods in Engineering*, 75(9):1085–1101, 2008.

[7] T. Rendall and C. Allen. Reduced Surface Point Selection Options for Efficient Mesh Deformation Using Radial Basis Functions. *Journal of Computational Physics*, 229(8):2810 – 2820, 2010.