Solvers for coupled sparse/dense FEM/BEM linear systems

Marek Felšöci

June 11, 2022

Contents

1 Introduction

This is an example experimental study relying on the `test_FEMBEM` solver test suite. Here, we are especially interested in solving coupled sparse/dense FEM/BEM linear systems arising in the domain of aeroacoustics. The idea is to evaluate the solvers available in the open-source version of `test_FEMBEM` for the solution of this kind of linear systems.

2 Experimental study

Unfortunately, the open-source version of `test_FEMBEM` [4] does not implement couplings of sparse and dense direct solvers which is normally our go-to method for solving sparse/dense FEM/BEM systems. Therefore, we rely here only on dense direct solvers, namely HMAT-OSS and Chameleon.

HMAT-OSS [3] is an open-source and sequential version of the compressed hierarchical $\mathcal{H}$-Matrix dense direct solver HMAT [7] developed at Airbus. Chameleon [1] is a fully open-source dense direct solver without compression.

As of the test case, we consider a simplified *short pipe* which is still close enough to real-life models (see Figure 1).

Note that all the benchmarks were conducted on a single quad-core Intel(R) Xeon(R) CPU W3520 @ 2.67GHz machine with Hyper-Threading and 8 GiB of RAM .

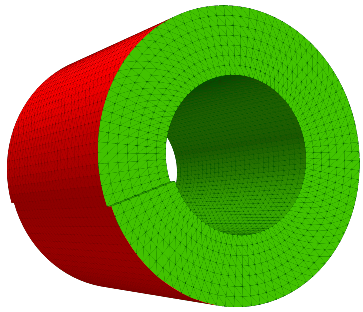

Figure 1: A *short pipe* mesh counting 20,000 vertices.

2.1 Data compression

In the first part, we want to know to which extent can data compression improve the computation time. For this, we compare sequential executions of both HMAT-OSS, the compressed solver, and Chameleon, the non-compressed solver, on coupled FEM/BEM systems of different sizes (see Figure 2). The results clearly show the advantage of using data compression, especially with increasing size of the target linear system.

For these experiments, we have considered the precision parameter ϵ for the HMAT-OSS solver to be 10^{-3}. In Figure 3, the relative error curve for the runs presented in Figure 2 verifies that the threshold is respected and that the error of the solutions computed by HMAT-OSS is even smaller than ϵ.

2.2 Multi-threaded execution

To study the impact of parallel execution on the time to solution, we limit ourselves to the Chameleon solver as HMAT-OSS is sequential-only. In Figure 4, we comapre the computation times of Chameleon on coupled FEM/BEM systems of different sizes using either one or four threads. According to the results, we can observe a significant decrease in computation time in case of parallel executions. Moreover the parallel efficiency of the run on the largest linear system considered (8000 unknowns) is approximately 79%.

3 Conclusion

We have evaluated the performance of the solvers branched to the `test_FEMBEM` test suite on coupled sparse/dense FEM/BEM linear systems. The solvers

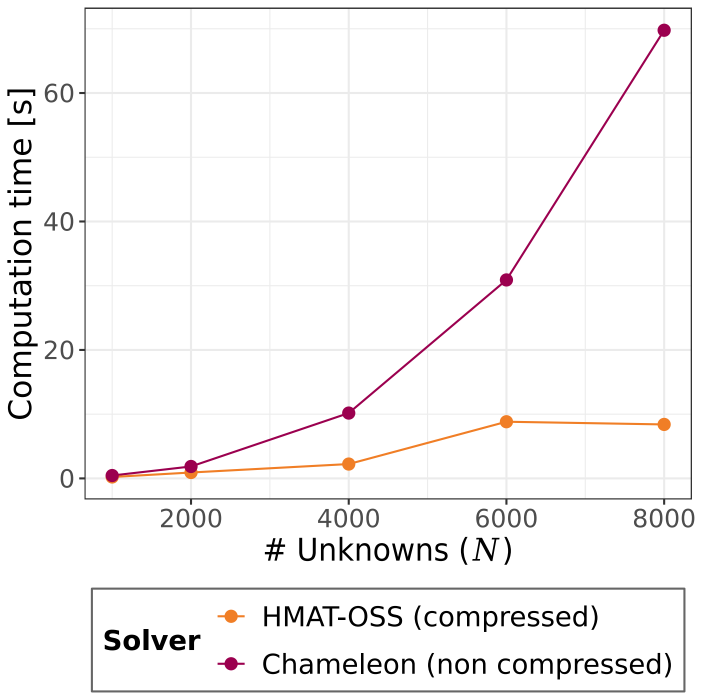

Figure 2: Computation times of sequential runs of HMAT-OSS and Chameleon on coupled sparse/dense FEM/BEM linear systems of varying size.

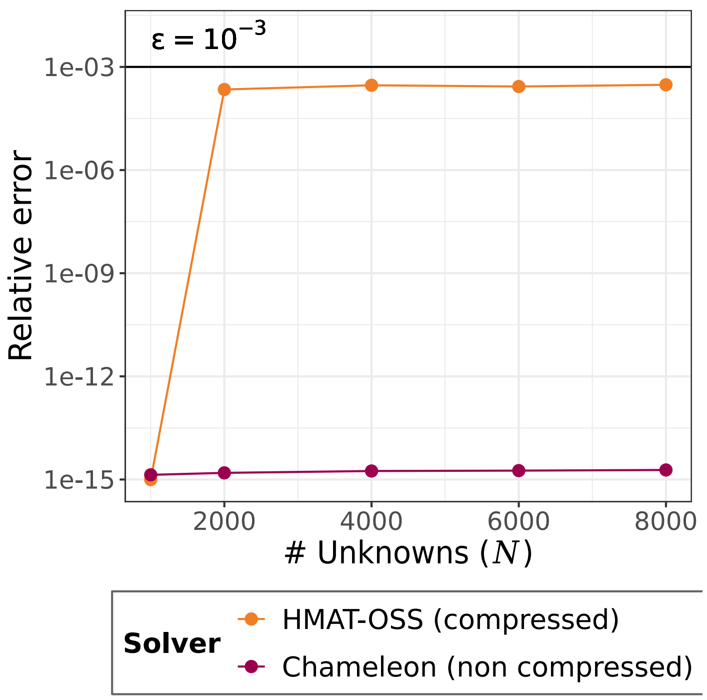

Figure 3: Relative error of sequential runs of HMAT-OSS and Chameleon on coupled sparse/dense FEM/BEM linear systems of varying size.

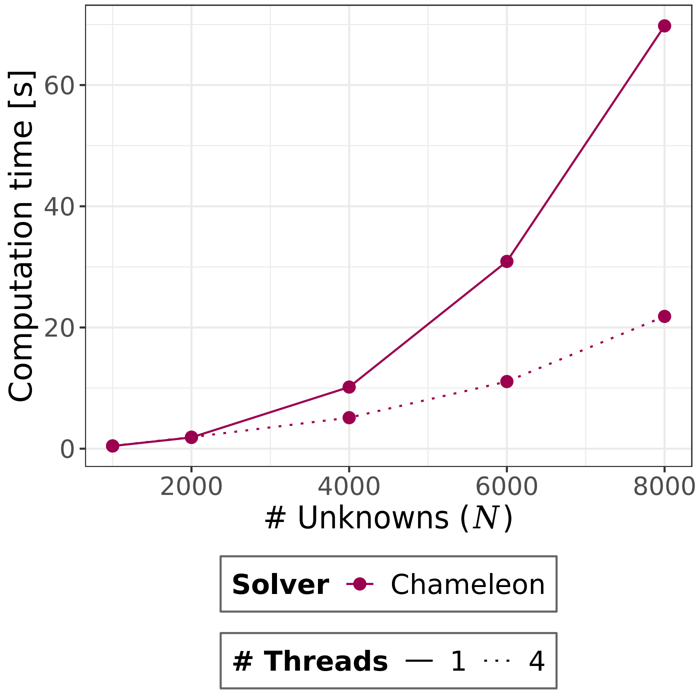

Figure 4: Computation times of sequential and parallel runs of Chameleon on coupled sparse/dense FEM/BEM linear systems of varying size.

considered were HMAT-OSS, a sequential compressed dense direct solver and Chameleon, a multi-threaded non-compressed dense direct solver.

The comparison of sequential runs of HMAT-OSS and Chameleon showed an important positive impact of data compression on the time to solution. In addition, the comparison of sequential and parallel runs of Chameleon as well as the computed parallel efficiency showed a considerable speed-up of the parallel execution.

4 Notes on reproducibility

With the aim of keeping the experimental environment of the study reproducible, we manage the associated software framework with the GNU Guix transactional package manager [2]. Moreover, relying on the principles of literate programming [6], we provide a full documentation on the construction process of the experimental environment, the execution of benchmarks, the collection and the visualization of results as well as on producing the final manuscripts in a dedicated technical report associated with this study [5]. A public companion contains all of the source code, guidelines and other material required for reproducing the study: `https://gitlab.inria.fr/tuto-techno-guix-hpc/test_fembem/advanced-setup`, archived on `https://archive.softwareheritage.org/` under the identifier `swh:1:snp:79f450e0f43828f56f261d81b3e86aaab18362eb`.

References

[1] *Chameleon, a dense linear algebra software for heterogeneous architectures.* https://gitlab.inria.fr/solverstack/chameleon.

[2] *GNU Guix software distribution and transactional package manager.* https://guix.gnu.org.

[3] *hmat-oss.* https://github.com/jeromerobert/hmat-oss.

[4] *test_FEMBEM, a simple application for testing dense and sparse solvers with pseudo-FEM or pseudo-BEM matrices.* https://gitlab.inria.fr/solverstack/test_fembem.

[5] M. Felšöci, *Solvers for coupled sparse/dense FEM/BEM linear systems: guidelines for reproducing the study*, Research Report RR-????, Inria Bordeaux Sud-Ouest, Feb. 3014.

[6] D. E. Knuth, *Literate Programming*, Comput. J., 27 (1984), p. 97–111.

[7] B. Lizé, *Résolution Directe Rapide pour les Éléments Finis de Frontière en Électromagnétisme et Acoustique : $\mathcal{H}$-Matrices. Parallélisme et Applications Industrielles.*, PhD thesis, Université Paris 13, 2014.