

Comparison of Methods for Large-Scale Sparse Matrix Inversion

Ivain Ignatenko and Pavel Oganessian

Abstract. This paper addresses the efficient inversion of large-scale sparse matrices. We review the Cholesky decomposition method, noting its challenges with density fill-in and parallel implementation. Our study compares the Eigen library's performance with the CHOLMOD module from SuiteSparse. Experimental results show that CHOLMOD significantly outperforms Eigen for large matrices.

Introduction

While solving modern computational tasks and spectral problems in particular, we are often faced with the need to compute the inverse of sparse symmetric matrices, predominantly containing zero elements.

In this work we concentrate on a special case of matrix inversion, which is a sub-problem within the implementation of an algorithm for finding eigenvalues of large-dimension sparse matrices. Such tasks are characterized by an exponential increase in computational complexity and associated escalating demands on computer memory.

1. Method

As an intermediate step for this sparse matrix inversion problem, we chose the Cholesky decomposition method. It represents a symmetric positive-definite matrix A in the form $A = LL^T$, where L is a lower triangular matrix with positive diagonal entries [1]. The method is more numerically stable and requires approximately half the number of arithmetic operations when compared to more general methods like Gaussian elimination or LU decomposition [1].

However, during the inversion process, the matrix can become filled with non-zero elements, up to $n^2/2$ in the worst case, where n is the matrix dimension.

This imposes additional constraints on the maximum dimension of the matrix for a fixed amount of computer memory [2].

The Cholesky decomposition method is inherently direct [3, 4], which makes the process of its parallel implementation significantly more complex [2]. Hence, an obvious solution is to use an existing mathematical library that offers an optimized multi-threaded implementation of the method for efficient large-dimension matrix inversion. However, not every library satisfies these requirements. In particular, many libraries do not support working with sparse matrices.

2. Results

Initially, we considered the Eigen library. Eigen is a C++ template library for linear algebra [5]. It implements two variants of self-adjoint (Hermitian) sparse matrix decomposition: $A = LL^T$ and $A = LDL^T$, respectively [6]. However, both implementations are unable to utilize the multi-threading capabilities [7] of the CPU, making them unsuitable for use with large-dimension matrices as shown in Table 1.

Matrix dimension (N)	Data size (MB)	Runtime (ms)
2187	1.2 MB	0.38
14739	11 MB	2904
27783	22 MB	14714
68208	56 MB	106779
107811	90 MB	399304

TABLE 1. Correlation between matrix dimension and `SimplicialLLT` algorithm runtime

The experimental results show that the algorithm’s execution time grows exponentially once the matrix data no longer fits entirely into the CPU cache. Intel® Core™ i7-12650H processor with 24 megabytes of Intel® Smart Cache [8] was used for the experiments.

As an alternative approach, we explored the `CHOLMOD` module from the SuiteSparse library. `CHOLMOD` is a set of routines for factorizing sparse symmetric positive definite matrices of the form A or AA^T , updating/downdating a sparse Cholesky factorization, solving linear systems, updating/downdating the solution to triangular system $Lx = b$, and many other sparse matrix functions for both symmetric and unsymmetric matrices [9].

The Cholesky decomposition implementation in the SuiteSparse library supports sparse matrices and their multi-threaded processing due to the utilization of OpenMP and CUDA technologies [10]. Table 2 shows the practical results obtained when processing large-dimension sparse matrices with `CHOLMOD` module procedures.

Comparison of Methods for Large-Scale Sparse Matrix Inversion

Matrix dimension (N)	Data size (MB)	Runtime (ms)
2187	1.2 MB	12
14739	11 MB	341
27783	22 MB	689
68208	56 MB	2267
107811	90 MB	4565
206763	185 MB	10158

TABLE 2. Correlation between matrix dimension and cholmod_factorize algorithm runtime

The graph of the experimental results presented in Figure 1 shows that the Cholesky decomposition implementation in the SuiteSparse library outperforms the analogous algorithm from Eigen library in terms of execution time when processing large-dimension matrices.

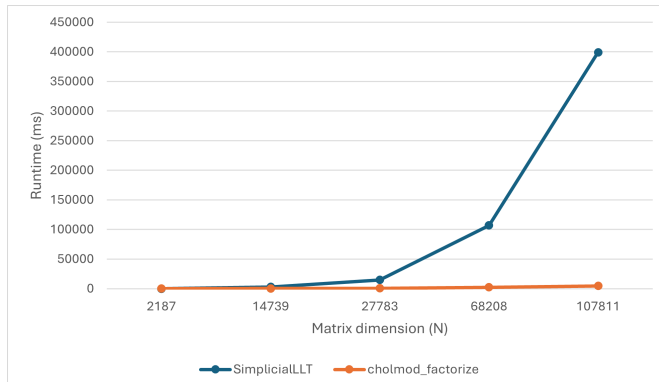


FIGURE 1. Comparison of the runtime of the cholmod_factorize and SimplicialLLT algorithms on the same input data

Conclusion

Thus, it can be concluded that the most effective solution of the large-dimension sparse matrix inversion method is the CHOLMOD module from the SuiteSparse library. Due to its multi-threading support, CHOLMOD demonstrates high performance, making it a preferred choice for computational tasks, including practical applications in spectral problems, including general eigenvalue problems. Future work will focus on comparison of such methods on modern GPU.

Acknowledgment

This research was supported by the Russian Science Foundation (project no. 22-11-00302-P).

References

- [1] Dereniowski, D. and Kubale M., *Cholesky Factorization of Matrices in Parallel and Ranking of Graphs*. Parallel Processing and Applied Mathematics. PPAM 2003. Lecture Notes in Computer Science, vol 3019. Springer, Berlin, Heidelberg. Available at: https://doi.org/10.1007/978-3-540-24669-5_127
- [2] Algowiki, *Cholesky decomposition*, 2025. Available at: https://algowiki-project.org/en/Cholesky_decomposition [Online; accessed 02-July-2025].
- [3] В.Р. Демидович and И.А. Марон, *Основы вычислительной математики*, Наука, 1970.
- [4] Д.К. Фаддеев and В.Н. Фаддеева, *Вычислительные методы линейной алгебры: Учебник для вузов*, Учебники для вузов. Специальная литература, Лань, 2002.
- [5] Eigen. <https://eigen.tuxfamily.org/index.php> [Online; accessed 02-July-2025].
- [6] Eigen, *SparseCholesky module*. https://eigen.tuxfamily.org/dox/group__SparseCholesky__Module.html [Online; accessed 02-July-2025].
- [7] Eigen, *Eigen and multi-threading*. <https://eigen.tuxfamily.org/dox/TopicMultiThreading.html> [Online; accessed 02-July-2025].
- [8] Wayback Machine, *Intel® Core™ i7-12650H Processor*. <https://web.archive.org/web/20250214094424/https://www.intel.com/content/www/us/en/products/sku/226066/intel-core-i712650h-processor-24m-cache-up-to-4-70-ghz/specifications.html> [Online; accessed 02-July-2025].
- [9] Timothy A. Davis, *User Guide for CHOLMOD: a sparse Cholesky factorization and modification package*, June 2025.
- [10] GitHub, *SuiteSparse*. <https://github.com/DrTimothyAldenDavis/SuiteSparse> [Online; accessed 02-July-2025].

Ignatenko. I. D.

Institute of Mathematics, Mechanics and Computer Science named after I.I. Vorovich
Southern Federal University
Rostov-on-Don, Russia
e-mail: iignatenko@sfedu.ru

Oganessian P. A.

MSU-BIT University
Shenzhen, China
Institute of Mathematics, Mechanics and Computer Science named after I.I. Vorovich
Southern Federal University
Rostov-on-Don, Russia
e-mail: poganessian@sfedu.ru