



String-Wave Direct Parallel Solver for Sparse System of Linear Equations

Alexey Y. Likhoshesterov^{1,2} , Yana G. Velikaya² 

© The Authors 2023. This paper is published with open access at SuperFri.org

The article discusses a parallel algorithm of solving linear algebraic equations systems for symmetric sparse matrices, which allows to split a large task into many small subtasks, thereby both increasing performance and reducing memory consumption. It is based on a method of simultaneous calculation of intermediate values during matrix factorization with maintaining load balancing on processors so that when the final result of the left parts of the factorization is obtained, the right parts of the factorization do not depend on them. This approach allows the initial stiffness matrix to be represented as a product of a large number of simple matrixes and solve a system of linear algebraic equations in the form of a sequence of solutions by substitution. To reduce the filling of sparse factorization matrices, an approximate minimum degree method was used, which, in addition to being one of the most efficient and fastest ones existing at the moment, allows the developed algorithm to distribute the load of calculations more evenly. The developed method is implemented in APM Ltd. software products for systems with shared memory, but it can also be performed for distributed memory systems.

Keywords: approximate minimum degree, string-wave algorithm, finite element method, system of linear equations, matrix factorization.

Introduction

Currently, the finite element method (FEM) is actively used for engineering calculations in the field of construction and machine-building [1]. The use of FEM involves solving systems of linear algebraic equations of large dimension. The matrices of such systems, called stiffness matrixes, have a number of properties that depend on the structure and number of nodes of the finite element grid, as well as the nature of the problem. These matrices are symmetric and sparse to a considerable extent.

With the help of the system of linear algebraic equations, such tasks as: static, dynamic calculation, calculation of natural frequencies, stability, nonlinear calculations, etc. are solved. In general, system of linear algebraic equations has the following form [2].

$$K_{n,n}X_{n,m} = F_{n,m}, \quad (1)$$

where K – is the stiffness matrix, X – are displacement vectors, F – are load vectors for a finite element system, n – is the dimension of the stiffness matrix; m – is the number of loads.

Due to the fact that the stiffness matrix is sparse, it is presented in the CSR (Compressed Sparse Rows) [3] string sparse form, where the storage structure uses three one-dimensional arrays: 1) an array of non-zero elements of the stiffness matrix line by line; 2) an array of column numbers of non-zero elements line by line; 3) an array of the location of the first non-zero element in each row. This format allows to reduce the amount of source data and decrease the number of operations performed on them significantly, i.e., when calculating, memory consumption is reduced and performance increases.

The article is organized as follows. Section 1 is devoted to the problem of factorization of symmetric sparse matrices and the need to reorder them. In section 2 we describe the main

¹APM Ltd., Korolev, Russian Federation

²Belgorod State University, Belgorod, Russian Federation

methods for reordering sparse matrices and compares the performance of the AMD and Cuthill–McKee algorithms. Section 3 contains a description of the developed string-wave algorithm for solving system of linear algebraic equations and computational experiments of its application. Conclusion summarizes the study and points directions for further work.

1. Matrix Factorization

Today, two different approaches to solving systems of linear algebraic equations are used: direct solution and iterative methods. Each approach has certain advantages, but direct methods are of the most common use. The main problem of iterative methods is the frequent poor conditionality of stiffness matrices of large dimension when its determinant is equal to a small value or tends to zero. And even the preconditioning process does not absolutely guarantee the convergence of the solution. In order to simplify the solution of system of linear algebraic equations with a symmetric stiffness matrix, the following factorization is used [4].

$$K_{n,n} = L_{n,n}D_{n,n}L_{n,n}^T, \quad (2)$$

where L – is the lower triangular matrix (stored in sparse format), in which all diagonal elements are equal to one, D – is the diagonal matrix (stored as a one-dimensional array).

The solution of system of linear algebraic equations $LDL^T X = F$ in this case is reduced to the sequential solution of three simplest issues by the substitution method.

$$\begin{cases} L_{n,n}Y_{n,m} = F_{n,m}; \\ D_{n,n}Z_{n,m} = Y_{n,m}; \\ L_{n,n}^T X_{n,m} = Z_{n,m}. \end{cases} \quad (3)$$

The main problem with this factorization is filling a sparse triangular matrix L with a large number of non-zero elements, which entails significant memory consumption and an increase in the number of computational operations on the elements of this matrix. Therefore, before the factorization procedure, the rows and columns of the original stiffness matrix are reordered. To do this, we find a special permutation matrix P such that

$$(P_{n,n}K_{n,n}P_{n,n}^T)(P_{n,n}X_{n,m}) = P_{n,n}F_{n,m}. \quad (4)$$

2. Permutation of Stiffness Matrix

The permutation matrix P is a matrix in which each row and column contains only one element equal to one, and the others equal to zero. This matrix is stored as an array of column numbers with single elements. To find the permutation matrix, the Cuthill–McKee algorithm is often used due to the simplicity of its implementation, but such an algorithm is not optimal from the point of reducing the filling of factorization matrices with nonzero elements. Algorithms based on the application of the minimum degree method are more effective. In practice, this method, in its original form, is not used due to its considerable complexity. Two of its modifications are widespread: the Multiple Minimum Degree method (MMD) and the Approximate Minimum Degree method (AMD) [5]. The latter is used in the software products of the company APM Ltd. Figure 1 shows the result of calculating the lower triangular matrix L with permutation using the Cuthill–McKee algorithm and the approximate minimum degree method.

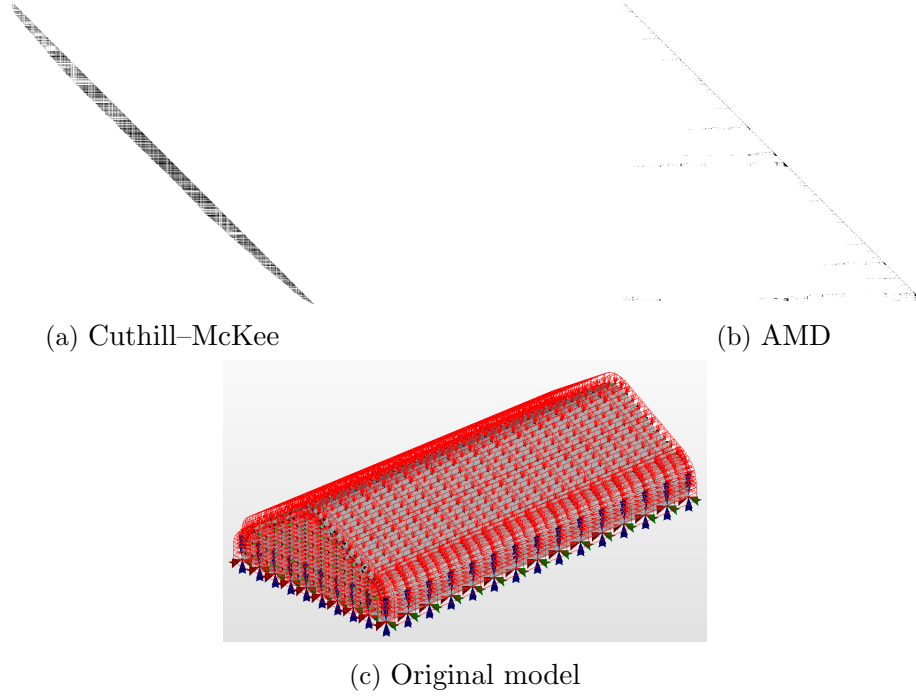


Figure 1. The result of factorization after reordering the matrix

Here the black dots denote the nonzero elements of the resulting factorization matrix L . Table 1 shows the number of received non-zero elements of this matrix.

Table 1. Results of matrix reordering algorithms

Initial stiffness matrix	Cuthill-McKee	AMD
279966	4510257	469879

After the permutation procedure, the factorization of the stiffness matrix is performed directly. In the classical version, LDL^T factorization calculated as follows

$$\begin{cases} D_j = K_{jj} - \sum_{k=1}^{j-1} L_{jk} D_k L_{jk}^T; \\ L_{ij} = \frac{K_{ij} - \sum_{k=1}^{j-1} L_{ik} D_k L_{jk}^T}{D_j}; \\ i = 1, \dots, n; j = 1, \dots, n. \end{cases} \quad (5)$$

3. Description of the Algorithm

Expression (5) does not lend itself well to parallelization due to the large number of information dependencies. In addition, the calculations are not balanced, which significantly reduces their efficiency [6]. To eliminate these problems, an algorithm adapted for parallel computing was developed. The triangular matrix of factorization L can be visually represented as a musical instrument like a harp (Fig. 2). Each harp string corresponds to a column of the matrix L . The strings are played alternately. Playing on the j -th string means finding the final values of the elements of the j -th column L . At the same time, each string propagates an undamped sound wave towards the strings located to the right of it. That is, each j -th column of the matrix L

partially changes the values of the elements of all columns with a number higher than j . Thus, it is possible to calculate the effect of the current column on all columns located to the right in parallel.

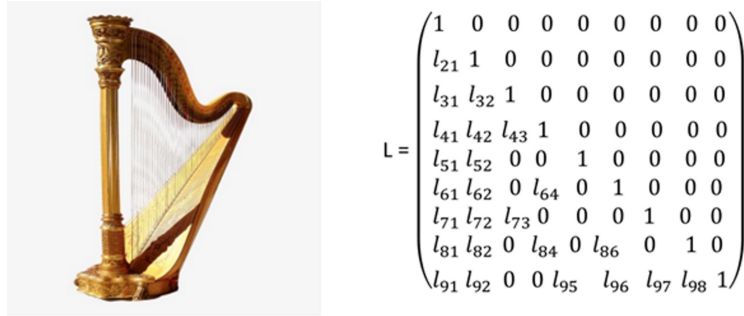


Figure 2. Visual representation of the factorization matrix in the form of a harp

Mathematically, the string-wave direct solver can be described as follows. First, the view matrices are assigned the corresponding values of the stiffness matrix.

$$\begin{cases} D_{ii} = K_{ii}; \\ L_{ij} = K_{ij}; \\ i = 1, \dots, i - 1; j = 1, \dots, i - 1. \end{cases} \quad (6)$$

Further, the elements of these matrices are changed with accordance to the algorithm described above.

$$\begin{cases} L_{ij} = \frac{L_{ij}}{D_{jj}}; \\ L_{ik} = L_{ik} - L_{kj}D_{jj}L_{ij}; \\ D_{ii} = D_{ii} - L_{ij}D_{jj}L_{ij}; \\ j = 1, \dots, n; i = j + 1, \dots, n; k = 2, \dots, i - 1. \end{cases} \quad (7)$$

Since the factorization matrix L is sparse, not all right columns are affected by the current one. In the course of the conducted research, it was revealed that only those columns which number corresponds to the row numbers of non-zero elements of the current column change. This circumstance makes it possible to eliminate unnecessary iterations of the parallel algorithm cycle. When solving system of linear algebraic equations of large dimensions by a direct method, there is often not enough memory. The string-wave algorithm allows to reduce the size of the problem, for example, by applying the following factorization.

$$K = L_1 B L_1^T, \quad (8)$$

where L_1 are the first columns of the triangular factorization matrix L , B is a submatrix of smaller dimension obtained by the influence of L_1 (Fig. 3). In turn, a similar factorization can

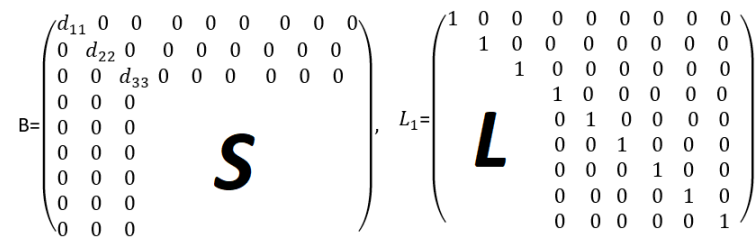


Figure 3. Reducing the dimension of the problem

be applied to the matrix B . As a result, the string-wave direct solver allows us to obtain the following expression

$$K = L_1 L_2 \dots L_{n-1} L_n D L_n^T L_{n-1}^T \dots L_2^T L_1^T. \quad (9)$$

This factorization of the stiffness matrix allows to split a large problem into many small subtasks, where the solution of system of linear algebraic equations is also found by substitution

$$\left\{ \begin{array}{l} L_1 X_n = F; \\ L_2 X_{n-1} = X_n; \\ \dots\dots\dots \\ L_n X_{i-1} = X_{i-2}; \\ DX_i = X_{i-1}; \\ L_n^T X_{i+1} = X_i; \\ \dots\dots\dots \\ L_2^T X_2 = X_3; \\ L_1^T X_1 = X_2. \end{array} \right. \quad (10)$$

Speedup and efficiency parameters were used to evaluate the efficiency of parallel computing [7]. Speedup refers to the execution time ratio of a sequential algorithm to the execution time of a parallel algorithm on p processors

$$S_p = \frac{T_1}{T_p}. \quad (11)$$

The efficiency of using processors by a parallel algorithm in solving a problem is defined as the ratio of acceleration to the number of processors

$$E_p = \frac{S_p}{p}. \quad (12)$$

The results of computational experiments evaluating the effectiveness of the parallel algorithm are presented in Tab. 2. The algorithm is implemented using OpenMP [8]. Calculations were performed on a six-core Intel Core i7-8700K processor with 32 GB of DDR4 RAM. To conduct computational experiments, the APM Structure3D module of APM WinMachine software was used, in which models of various dimensions were loaded and when measuring the execution time of the parallel algorithm, the time for the formation of the stiffness matrix was taken into account.

Table 2. Evaluating the effectiveness of a parallel algorithm

Dimension of the stiffness matrix, n	Speedup, S_p	Efficiency, E_p
94,692	3.54	0.59
174,267	3.75	0.63
244,620	4.86	0.81
527,919	4.72	0.79
755,181	4.78	0.80

Thereby, the string-wave direct solver effectively works in conjunction with the AMD re-ordering algorithm, breaking a large task into many small subtasks with the least information dependencies, which allows its parallel implementation to achieve optimal values of speedup and efficiency parameters.

Conclusion

The article presents a new algorithm for solving systems of linear algebraic equations for symmetric sparse matrices. This algorithm includes the AMD method for reordering sparse matrices and the string-wave method for solving system of linear algebraic equations of our own design. It is successfully used in software products of the company APM Ltd. The developed solver is used in such software modules as static and dynamic calculations, calculation of natural frequencies and stability when solving a generalized eigenvalue problem, calculation of harmonic vibrations and many others.

The developed algorithm is parallel and implemented using OpenMP technology. Computational experiments show that with an increase in the number of processors and the volume of input data, the speedup increases without a decrease in efficiency. That is the developed parallel algorithm is scalable.

Further research will focus on improving its performance and implementation using CUDA technology for NVIDIA graphics accelerators and MPI technology for distributed memory systems.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Bathe, K.J., Wilson, E.L.: Numerical methods in finite element analysis. Stroyizdat, Moscow (1982).
2. Demmel, J.: Computational linear algebra. Theory and applications. Word, Moscow (2001).
3. Pissanetsky, S.: Sparse Matrix Technology. Academic Press Inc (1984).
4. Andreev, B.: Numerical methods. Publishing department of the Faculty of Computational Mathematics and Mathematics of Lomonosov Moscow State University, Moscow (2013).
5. Heggenes, P., Eisenstat, S.C., Kumfert, G.: The Computational Complexity of the Minimum Degree Algorithm. In: NASA/CR-2001-211421ICASE Report No. 2001-4, Langley Research Center, Hampton, Virginia (2001). <https://doi.org/10.2172/15002765>
6. Voevodin, V., Voevodin, Vl.: Parallel Computing. BHV, Saint Petersburg (2002).
7. Gergel, V.P.: High-performance computing for multiprocessor, multi-core systems. Moscow University Publishing House, Moscow (2010).
8. Gergel, V.P.: Modern languages and parallel programming technologies. Moscow University Publishing House, Moscow (2012).