CYBERNETICS
and COMPUTER
TECHNOLOGIES

UDC 519.6
DOI:10.34229/2707-451X.25.3.6

V. SYDORUK, O. CHYSTIAKOV, P. BENNER, O. NIKOLAIEVSKA

SOLVING LINEAR SYSTEMS OF EQUATIONS WITH BLOCK SKYSCRAPER STRUCTURE

This paper presents an efficient parallel algorithm for solving systems of linear algebraic equations (SLAE) with block skyscraper matrices on shared-memory systems. The algorithm is optimized for high performance and scalability through effective workload distribution among processor cores. It was tested on a modern computational cluster by solving practical structural strength modeling problems. Performance analysis covered execution time, speedup, and the influence of block size. Results show significant time reduction and stable scalability with increasing core count, confirming the method's suitability for large-scale engineering simulations in structural mechanics, applied physics, and other computationally intensive fields.

Keywords: *high-performance computing, system of linear algebraic equations, Cholesky method, sparse matrices, block skyscraper matrices, parallel algorithms, systems with shared memory.*

Introduction. During the numerical solution of scientific and technical problems, in many cases there is a need to solve systems of linear algebraic equations (SLAE). Moreover, as a rule, the solution of SLAE takes a significant part (75 % or more) of the time of solving the entire problem in general. For example, SLAE arise as a result of the discretization of boundary value problems or eigenvalue problems by the projection-difference method (finite differences, finite elements). Also, solving of SLAE often is the bottleneck in algorithms for solving systems of nonlinear equations, integrating systems of differential equations, etc.

An important feature of linear algebra problems that arise as a result of discretization is that the number of nonzero matrix elements of such problems is kn, where $k \ll n$, and n is the order of the matrix, that is, the matrices are sparse. The structure of the sparse matrix is determined by the numbering of the unknowns of the problem and is often band, profile, block-diagonal with framing, etc. Another feature of matrices is their size, which can reach tens of millions. This is determined by the desire to work with more accurate and detailed models of processes and objects.

The development of methods for solving SLAE with sparse matrices is covered in the monographs of A. George and J. Liu, O. Esterbu, M. Zlatov, S. Pissanetzky, Y. Saad, J. Golub, R.P. Tewarson, M.Yu. Balandin and others. The monograph [1] highlights a number of fundamental principles of working with sparse matrices. In the monograph [2], a wide range of issues related to both the methods of solving SLAE and auxiliary problems, such as data storage formats or preordering optimization algorithms, are considered.

The works of H. Alan, J.M. Ortega, J. Dongarra, I.M. Molchanov, V.V. Voevodin, V.V. Voevodin, O.M. Khimich and others (for example, [3–8]) are devoted to the development of parallel algorithms for solving systems with sparse matrices. The monograph [9] considers the construction and research of parallel algorithms for solving linear algebra problems on parallel computer systems of a certain architecture.

In [10], a multi-threaded solver for symmetric positive definite linear systems where the coefficient matrix of the problem features a bordered-band non-zero pattern is presented. Various aspects of parallel algorithms are discussed in papers [11–17]. In [18], algorithmic software for mathematical modeling of structural stability is considered, which is reduced to solving a partial generalized eigenvalues problem of sparse matrices, with automatic parallelization of calculations on modern parallel computers with graphics processors. In [19], the integrated approach is proposed for the numerical prediction of the strength and reliability of pipelines with detected defects in corrosion loss of metal with spatially inhomogeneous properties using the Monte Carlo method to implement a statistical analysis of strength alongside a finite element description of the state of a specific structure. The perturbation theory and properties of SLAE with approximate initial data and the sensitivity of the solution of the WLS problem with approximate initial data is considered in [20].

Over the past decades, a number of libraries have been created based on the developed algorithms, which include programs for solving SLAE with sparse matrices: SparseBLAS [21], IMSL [22], NAG [23], and others. Among the software tools designed for parallel computers, effective implementations of algorithms with sparse matrices are offered by Hypre [24], MUMPS [25], SUPERLU [26], SPARSKIT [27] libraries.

The solution of SLAE with sparse matrices of large orders requires significant computing resources, which can be provided by modern high-performance computing systems, in particular, computers of MIMD architecture and computers of hybrid architecture (multicore CPU and GPU). Therefore, it is relevant to create effective parallel algorithms for solving linear algebra problems with sparse matrices. The effectiveness of parallel algorithms largely depends on the balance of loading processors, which in solving linear algebra problems is largely determined by methods of distribution between processes, storage and processing of matrices and vectors of the solved problem. At the same time, it is necessary to strive for uniform loading at every moment of time of all processors involved in solving the problem, minimizing the time the processors are in the waiting state. This is achieved by structural regularization of the matrix, in particular by the method of minimum degree. As a rule, as a result of applying such structural regularization, we get a matrix of a “skyscraper” structure. To effectively use the architectural features of modern hardware and software solutions in the field of high-performance computing, it is necessary to develop block algorithms.

In this note, a parallel algorithm for solving systems of equations with symmetric nonsingular block skyscraper matrices on a computer with shared memory is proposed. These matrices arise, in particular, in the problems of mathematical modeling in the construction industry (linear problems of the statics and dynamics of building structures). The results of testing when solving practical problems at nodes of a compute cluster are presented.

1. Solution of SLAE with block skyscraper matrices

1.1. Formulation of the problem

The problem of solving a system of linear algebraic equations is considered

$$Ax = b \tag{1}$$

where A is a square matrix of order n, b is a vector of the right-hand part of dimension n and x is a vector of solutions of dimension n.

Here, it is assumed that the matrix A resulting from the discretization of the original problem has large dimensions and is nonsingular, symmetric, and sparse.

1.2. Scheme of storage of block skyscraper matrices

When constructing parallel algorithms for solving SLAE with sparse matrices, the choice of the method of storing non-zero elements is of particular importance. However, data storage schemes of a general type, for example, a sparse row format, do not take into account the specifics of the matrix portrait. The use of information about the features of the matrix portrait when building a scheme for storing non-zero elements makes it possible to speed up access to matrix elements and perform operations on them. One such example is the “skyscraper” scheme [2, 4].

Often in practice, for example, when calculating building constructions, there are problems with symmetric matrices of a characteristic structure (fig. 1). In the upper triangular matrix of a LU-type decomposition like LDL^T, the non-zero values of the elements are placed in vertical bands of several adjacent columns, which look like skyscrapers. Thus, any row of the upper triangle consists of consecutive groups of nonzero and zero elements. Since non-zero elements are stored in groups, there is no need to store indices of each element, which makes it possible to reduce the dimension of index arrays. This is the essence of the skyscrapers scheme for storing matrix elements. The considered scheme is a modification of the sparse row format. Such a structure, in addition to saving memory, creates a prerequisite for using blocks. The elements in the blocks are stored in a column format, which matches well with BLAS routines that implement the corresponding matrix-matrix operations.

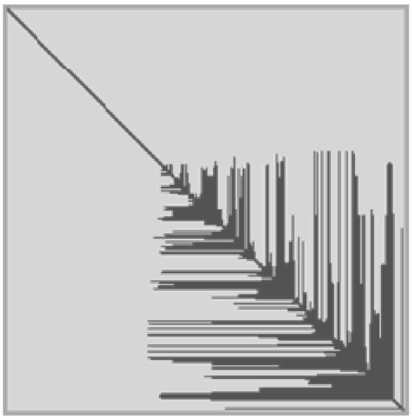

FIG. 1. An example of a skyscraper matrix

1.3. Algorithm for solving SLAE with block skyscraper matrices

It is known [9, 28] that for solving SLAE

$$Ax = b$$

with a symmetric positive definite matrix, the most effective direct method is the Choletsky method, which performs the LDL^T-decomposition of A, where L is a lower triangular matrix with 1 on the main diagonal, and D is a diagonal matrix.

Therefore, finding a solution to system (1) is reduced to decomposition

$$A = LDL^T \tag{2}$$

and the following solution of two triangular systems

$$Ly = b, \tag{3}$$

$$DL^T x = y. \tag{4}$$

Since the largest number of operations occurs precisely at the stage of matrix decomposition, a block parallel algorithm is considered, which takes into account the block-sparse structure of matrices and uses mainly matrix-matrix operations. To solve systems (3), (4), it is advisable to use existing programs of the MKL library for solving triangular systems (mkl_dcsrtrsv) with several right-hand side vectors, the matrices of which are presented in a CSR data format (see, for example, [29]).

1.4. A parallel algorithm for solving a linear system

Consider the case of matrix decomposition of a block-sparse structure on a computer with p CPUs and shared memory.

Let A be a square matrix of order n. Let's divide the matrix A into blocks of size $s \times s$. Grid sizes of blocks is , where $N = \left[\frac{n}{s}\right]$, under the condition that n is divisible by s, otherwise $N = \left(\left[\frac{n}{s}\right] + 1\right)$. The view of the lower triangular matrix of the block skyscraper structure is shown in fig. 2. The columns of the blocks of the lower triangle are cyclically distributed among the processors.

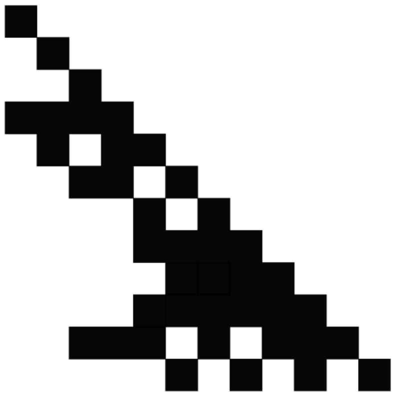

FIG. 2. View of the lower triangular matrix of the block skyscraper structure

On the K-th step ($K = 1, 2, \ldots, N$) the submatrix $A_R^{(K-1)}$ is processed. This is a square submatrix of order $R + s = n - Ks + s$, ($A_R^{(0)} \equiv A$), which is located at the bottom right of the output matrix.

Let us denote the matrices of the order $R + s$: $L^{(K)} = \begin{pmatrix} L_{11}^{(K)} & 0 \\ L_{21}^{(K)} & I_R \end{pmatrix}$ is lower triangular, $D^{(K)} = $ $= \begin{pmatrix} D_1^{(K)} & 0 \\ 0 & A_R^{(K)} \end{pmatrix}$ is diagonal, where $L_{11}^{(K)}$ is lower triangular block with 1 on the main diagonal, $D_1^{(K)}$ is diagonal block of order s, $L_{21}^{(K)}$ is rectangular submatrix of size R×s, that preserves the sparse structure of the block $A_{21}^{(K)}$, I_R is a unit submatrix of order R.

Then $A_{12}^{(K)} = \left(A_{21}^{(K)}\right)^T$

$$A_R^{(K-1)} = \begin{pmatrix} A_{11}^{(K)} & A_{12}^{(K)} \\ A_{21}^{(K)} & A_{22}^{(K)} \end{pmatrix} = \begin{pmatrix} L_{11}^{(K)} & 0 \\ L_{21}^{(K)} & I_R \end{pmatrix} \begin{pmatrix} D_1^{(K)} & 0 \\ 0 & A_R^{(K)} \end{pmatrix} \begin{pmatrix} L_{11}^{(K)} & 0 \\ L_{21}^{(K)} & I_R \end{pmatrix}^T.$$

Calculations at the K-th step are carried out in the following sequence:

- LDL^T block decomposition

$$A_{11}^{(K)} = L_{11}^{(K)} U_{11}^{(K)}, \tag{5}$$

where $U_{11}^{(K)} = D_{11}^{(K)} \left(L_{11}^{(K)}\right)^T$ is a temporary array only for the K-th step of decomposition;

- calculation of blocks $L_{21}^{(K)}$, $U_{12}^{(K)}$ (solving of the matrix SLAE)

$$U_{12}^{(K)} = \left(L_{11}^{(K)}\right)^{-1} A_{12}^{(K)}, L_{21}^{(K)} = \left(D_{11}^{(K)}\right)^{-1} \left(U_{12}^{(K)}\right)^T; \tag{6}$$

- s-rank modification of the submatrix $A_{22}^{(K)}$ (submatrix calculation $A_R^{(K)}$)

$$A_R^{(K)} = A_{22}^{(K)} - L_{11}^{(K)} U_{12}^{(K)}. \tag{7}$$

The parallel block algorithm can be described as follows.

For K ($K = 1, 2, \ldots, N-1$) we perform the following macro operations:

- factorization of the diagonal block according to (5); to implement the stage, we use the implementation of the parallel algorithm for the decomposition of a dense matrix;
- in all streams, we calculate the value of the blocks $L_{21}^{(K)}, U_{12}^{(K)}$ according to (6); we use the construction #pragma omp parallel for (static: 2) [11];
- in all streams, the s-rank modification of the submatrix $A_R^{(K)}$ is carried out according to (7), we use the construction #pragma omp parallel for (dynamic: 2) [30].

For $K = N$, we factorize the diagonal block $A_{11}^{(N)}$.

Note. Operations (7) are performed cyclically in all threads. Moreover, calculations are performed only with matrix blocks $A_{22}^{(K)}$ for which non-zero blocks from matrices $L_{21}^{(K)}, U_{12}^{(K)}$ have been calculated.

2. Verification and application of the algorithm

On the basis of the proposed parallel algorithm, software modules were created, which were tested when solving the problems of calculating the strength of building structures. The following tasks were chosen for the tests: analysis of the stressed-deformed state of the foundation of the building, static calculation of the stressed-deformed state of a 27-story residential building and analysis of the structural strength of a multistory office center.

Testing was carried out on the nodes of the SKIT-5 8-node cluster [31] with a peak performance of 100 TFlops. Each node is built on the basis of two 96-thread AMD EPYC 7642 processors, has 256 GB of RAM, integrated with the total data storage of the cluster complex with a volume of 200 TB. As operating system, Ubuntu 20.04 was used.

For the software implementation of the algorithm, the following OpenMP technology was used to organize parallelism between streams and the distribution of blocks of skyscrapers to the corresponding streams; modules of the Intel® MathKernelLibrary (Intel® MKL) 2019 library for efficient matrix-matrix operations.

2.1. Analysis of the stress-strain state of the foundation of the building

The general view of the structure and the used finite-element mesh are shown in fig. 3 on the left. The structure is divided into 94,346 finite elements. The corresponding finite element mesh contains 97,412 nodes. As a result of the discretization, the SLAE of the order of 283,031 was obtained, the portrait of the upper matrix of decomposition is shown in fig. 3 on the right.

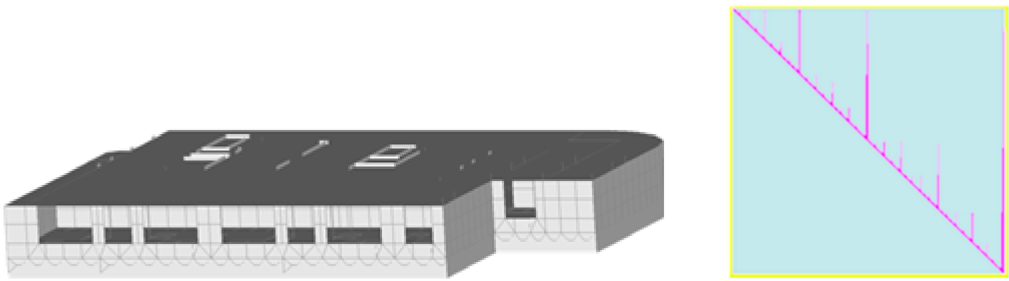

FIG. 3. Construction of the foundation and a portrait of the corresponding upper triangular matrix of the decomposition

The graphs of times and accelerations obtained when solving the problem with different number of threads using the block size of 160 are shown on fig. 4.

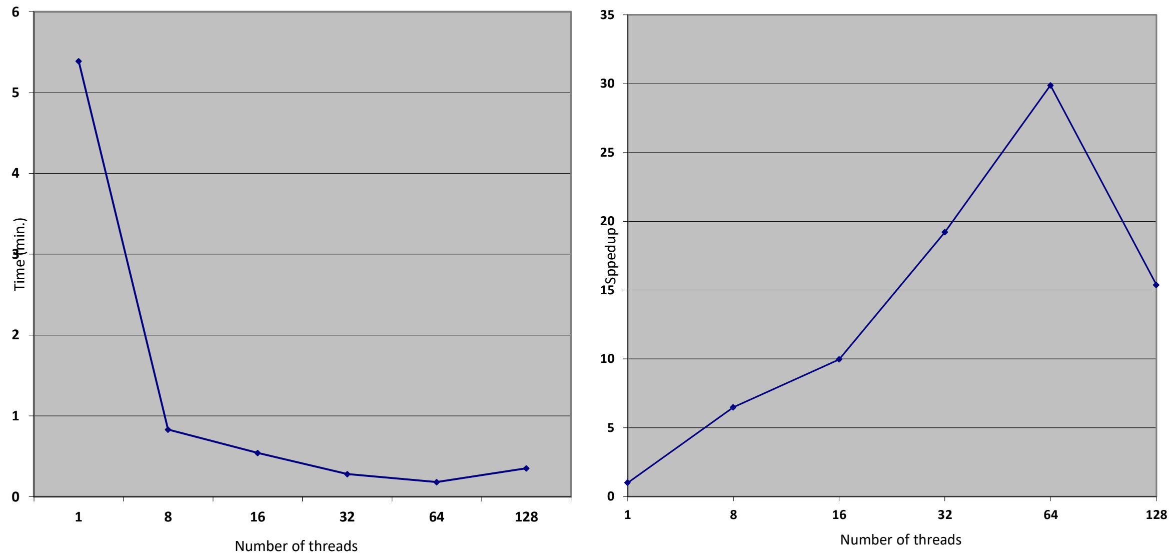

FIG. 4. Dependencies of time and acceleration on the number of threads for SLAE of the order of 283,031

2.2. Static calculation of the stress-strain state of a 27-story residential building

The general view of the structure of a residential building and the finite element grid used to study its structural stability are shown in fig. 5 on the left. The structure is divided into 121,761 finite elements. The corresponding finite element mesh contains 110,265 nodes. As a result of the discretization, an SLAE of order 661,590 was obtained, the portrait of the upper triangular matrix of its decomposition is shown in fig. 5 on the right. The graphs of the factorization time and accelerations obtained when solving the problem with different numbers of threads using a block size of 256 are shown in fig. 6.

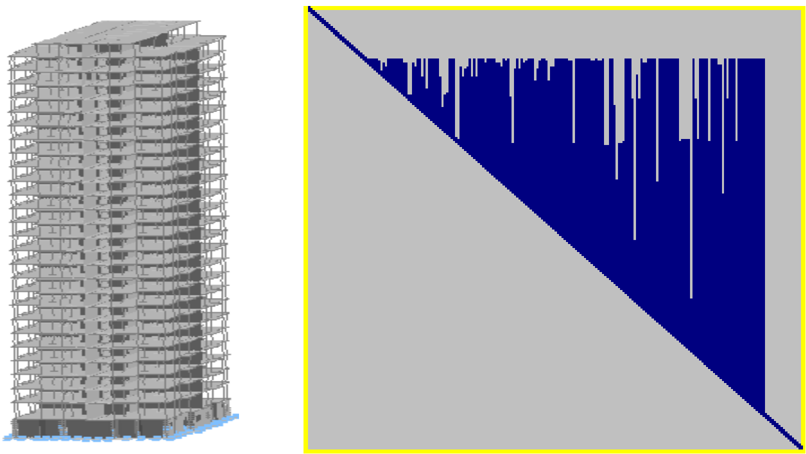

FIG. 5. Construction of a 27-story residential building and a portrait of the corresponding upper triangular matrix of its decomposition

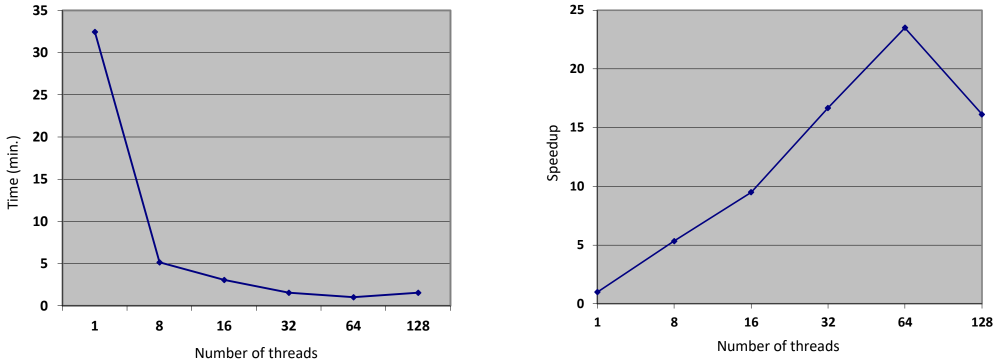

FIG. 6. Dependencies of time and acceleration on the number of threads for SLAE of the order of 661,590

2.3. Static calculation of the stress-strain state of a multi-story office center.

The general view of the construction of a multi-story office center and the finite element mesh used are shown in fig. 7 on the left. The structure is divided into 139,839 finite elements. The corresponding finite element mesh contains 124,490 nodes. As a result of the discretization, an SLAE of order 746,937 was obtained, the portrait of the upper triangular matrix of its decomposition is shown in fig. 7 on the right. The graphs of factorization time and accelerations obtained when solving the problem on different number of threads using block size 256 are shown in fig. 8.

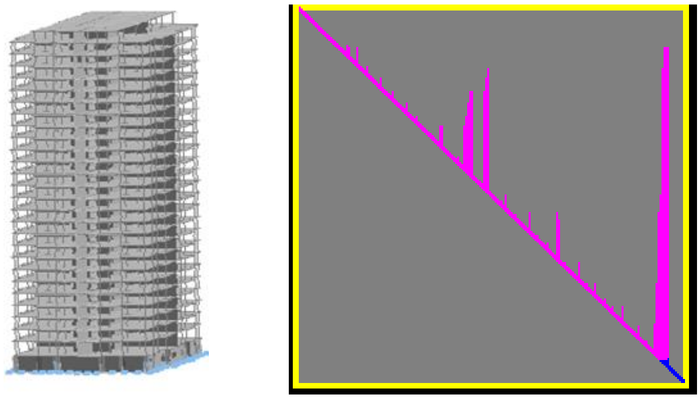

FIG. 7. Construction of a multi-story office center and a portrait of the corresponding upper triangular matrix

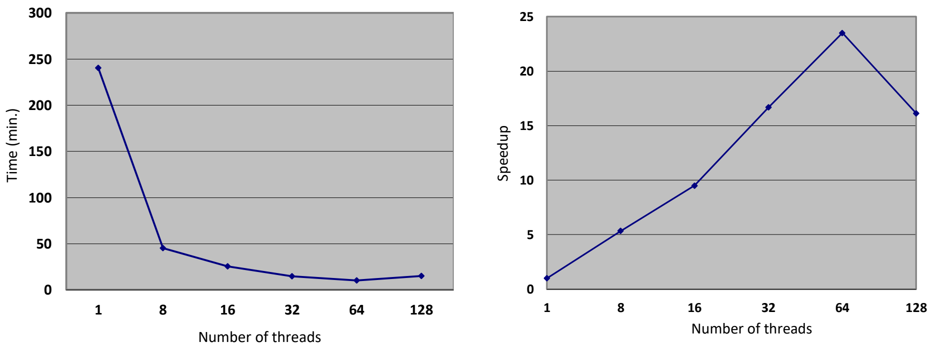

FIG. 8. Dependencies of time and acceleration on the number of threads for SLAE of the order of 661,590

Table shows the results of testing the parallel algorithm on several SLAEs of different orders with sparse matrices of the skyscraper structure.

TABLE. Time (minutes) of program operation for different number of threads

Task	Count of equa-tions	Count of non-zero elements	1	8	16	32	64	128
1	283,031	4 306,879	5.38	0.83	0.54	0.28	0.18	0.35
2	661,590	11,673,464	32.45	5.17	3.08	1.57	1.01	1.55
3	746,937	15,039,703	240.45	45.22	25.31	14.83	10.23	14.9

The results presented in fig. 4, 6, 8, and in Table demonstrate the dependence of algorithm characteristics (time to solve the problem, acceleration) on the balance of process loading. The conducted studies indicate that the size of the block with which the calculations are performed has a significant impact on the acceleration of calculations, as it affects both the speed of sequential operations and the number of operations that can be performed in parallel. An important role is also played by filling the columns of the block skyscraper matrix with non-zero blocks, because it actually determines how many operations are performed in parallel.

The strong scaling is demonstrated in our graphs and table for a fixed problem size and a variable number of processors p. The overhead shown in the graphs is explained by the effect of Amdahl's law, the costs of caching calculations and the different structure of skyscrapers in each core for different p. The graphs demonstrate the good scalability of the algorithm up to 64 threads, after which the block-sizes become too small.

As for weak scaling, when the size of the load on each processor remains constant, this case is interesting in theory, but in this case it is difficult to implement for a skyscraper structure. Weak scalability can really be demonstrated in the case of a matrix of a regular structure, for example, a strip (if the width of the strip is preserved).

Conclusions. For problems with sparse symmetric matrices of block skyscraper structure, a parallel implementation of a direct factorization-based method is proposed, which ensures high efficiency of parallelization, takes into account the structure of sparse matrices and their filling with data. The developed algorithm makes it possible to distribute among the processes the calculation with blocks of non-zero elements of the triangular decomposition of the sparse matrix in such a way that they are carried out simultaneously by the majority of processes.

Testing of the proposed algorithm was carried out when solving the problems of modeling the strength of building structures. The obtained results testify to the good scalability and efficiency of the proposed algorithm. These results are the basis for choosing an effective block size and the number of threads for solving structural strength modeling problems on SKIT 5 computing nodes.

It is advisable to direct further research to the search for algorithms for rearranging sparse matrices, which would ensure the highest efficiency of parallel algorithms for solving linear algebra problems, primarily due to the balanced loading of processes.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Max-Planck-Gesellschaft zur Förderung der Wissenschaften e.V. (Hofgartenstraße 8, 80539 München, Germany) via the EIRENE, a Max PlanckUkraine Cooperation and Mobility Grant, agreement from May 02, 2023.

Authorship contribution: Volodymyr Sydoruk & Oleksii Chystiakov – Research and development of the article, formulation of the research problem, writing the manuscript, proposing future research directions. Peter Benner – Formulation of the research problem, analysis of the obtained results, editing the manuscript. Olena Nikolaievska – Research and development of the article, analysis of the obtained results, writing the manuscript.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Acknowledgements

We are thankful to the Max-Planck-Gesellschaft zur Förderung der Wissenschaften e.V. for supporting this research via an EIRENE grant.

References

1. Pissanetzky S. *Sparse Matrix Technology*. Academic Press, 2008. https://doi.org/10.1016/C2013-0-11311-6
2. George A., Liu J. Numerical solution of large sparse systems of equations. M.: Mir, 1984. 334 p. (Russian)
3. Dongarra J., Beckman P., Moore T. et al. The International Exascale Software Project roadmap. *The International Journal of High Performance Computing Applications*. 2011. Vol. 25, Iss. 1. P: 3–60. https://doi.org/10.1177/1094342010391989
4. Khimich A.N., Popov A.V., Polyanko V.V. Algorithms of parallel computations for linear algebra problems with irregularly structured matrices. *Cybern Syst Anal*. 2011. Vol. 47. P. 973–985. https://doi.org/10.1007/s10559-011-9377-4
5. Baranov A.Yu., Popov A.V., Slobodyan Y.E., Khimich A.N. Mathematical modeling of building constructions using hybrid computing systems. *J. Autom. Inform. Sci.* 2017. **49** (7). P. 18–32. https://doi.org/10.1615/JAutomatInfScien.v49.i7.20
6. Khimich A.N., Popov A.V., Chistyakov O.V. Hybrid algorithms for solving the algebraic eigenvalue problem with sparse matrices. *Cybern Syst Anal*. 2017. **53** (6). P. 937–949. https://doi.org/10.1007/s10559-017-9996-5
7. Khimich O.M., Popov O.V., Chistyakov O.V. et al. A Parallel Algorithm for Solving a Partial Eigenvalue Problem for Block-Diagonal Bordered Matrices. *Cybern Syst Anal***.** 2020. **56**. P. 913–92. https://doi.org/10.1007/s10559-020-00311-z
8. Khimich O.M., Popov A.V., Chystiakov O.V. About the Peculiarities of Using Sparse Matrices in Problems 0f Mathematical Modeling. *Programming problems*. 2022. Vol. 3-4. P. 240–248. https://doi.org/10.15407/pp2022.03-04.240
9. Khimich A.N., Molchanov I.N., Popov A.V., Chistyakova T.V., Yakovlev M.F. *Parallel Algorithms to Solve Problems in Calculus Mathematics*. Kyiv: Nauk.Dumka, 2008. (in Russian)
10. Benner P., Ezzatti P., Quintana-Ortí E.S., Remón A. A Parallel Multi-threaded Solver for Symmetric Positive Definite Bordered-Band Linear Systems. In: Wyrzykowski R., Deelman E., Dongarra J., Karczewski K., Kitowski J., Wiatr K. (eds), Parallel Processing and Applied Mathematics. PPAM 2015. Lecture Notes in Computer Science. 2016. Vol 9573. Springer, Cham. https://doi.org/10.1007/978-3-319-32149-3_10
11. Sydoruk V., Yershov P. Adaptive algorithm for solving systems of equations with block-skyscraper matrices. *International Scientific Technical Journal "Problems of Control and Informatics"*. 2023. **67** (5). P. 17–31. https://doi.org/10.34229/2786-6505-2022-5-2
12. Khimich O.M., Popov O.V., Chistyakov O.V., Kokhanovsky V.O. Adaptive Algorithms for Solving Eigenvalue Problems in the Variable Computer Environment of Supercomputers. *Cybern Syst Anal*. 2023. Vol. 59. P. 480–492. https://doi.org/10.1007/s10559-023-00583-1
13. Khimich O.M., Popov A.V. Solving Ill-Posed Problems of the Theory of Elasticity Using High-Performance Computing Systems. *Cybern Syst Anal*. 2023. **59** (5). P. 743–752. https://doi.org/10.1007/s10559-023-00610-1
14. Petryk M.R., Boyko I.V., Khimich O.M., Sydoruk V.A. High-throughput methods for modeling nanoabsorption and diffusion with feedback in heterogeneous cylindrical multicomponent nanoporous media. *Cybern Sys tAnal*. 2023. **59** (6). P. 163–177. https://doi.org/10.1007/s10559-023-00636-5
15. Khimich O., Popov A., Chistyakov A. Effective Use of Sparse Matrices in Problems of Mathemaical Modeling. *CEUR Workshop Proceedings (CEUR-WS.org)*. 2023. https://ceur-ws.org/Vol-3501/s20.pdf

16. Khimich A., Polyanko V., Chistyakova T. Parallel algorithms for solving linear systems on hybrid computers. *Cybernetics and Computer Technologies*. 2020. 2. P. 53–66. https://doi.org/10.34229/2707-451X.20.2.6
17. Khimich O.M., Sydoruk V.A., Nesterenko A.N. Hybrid algorithm Newton method for solving systems of nonlinear equations with block Jacobi matrix. *Problems in programming*. 2020. Vol. 2-3. P. 208–217. https://doi.org/10.15407/pp2020.02-03.208
18. Chistyakov A.V. On improving the efficiency of mathematical modeling of the problem of stability of construction. *Artificial Intelligence*. 2020. **25** (3). P. 27–36. https://doi.org/10.15407/jai2020.03.027
19. Velikoivanenko E.A., Milenin A.S., Popov A.V., Sydoruk V.A., Khimich A.N. High-performance methods for analyzing the statistical strength of welded pipelines and pressure vessels using the monte carlo method. *Journal of Automation and Information Sciences*. 2020. **52** (11). P. 12–27. https://doi.org/10.1615/JAutomatInfScien.v52.i11.20
20. Khimich A.N., Nikolaevskaya E.A., Baranov I.A. Weighted least squares perturbation theory. Edited by Dr. Mykhaylo Andriychuk «*Matrix Theory – Classics and Advances*», 2023. https://doi.org/10.5772/intechopen.102885
21. SparseBLAS: https://netlib.org/sparse-blas/index.html (accessed: 11.08.2025)
22. IMSL: https://www.imsl.com/ (accessed: 11.08.2025)
23. NAG: https://nag.com/nag-library/ (accessed: 11.08.2025)
24. Hypre: https://hypre.readthedocs.io/en/latest/ (accessed: 11.08.2025)
25. MUMPS: https://mumps-solver.org/index.php (accessed: 11.08.2025)
26. SUPERLU: https://portal.nersc.gov/project/sparse/superlu/ (accessed: 11.08.2025)
27. SPARSKIT:https:/www-users.cse.umn.edu/~saad/software/SPARSKIT/ (accessed: 11.08.2025)
28. Molchanov I.N. Machine methods for solving problems of applied mathematics. Algebra, approximation of functions, ordinary differential equations. Kyiv: Nauk.Dumka, 2007. 550 p. (in Russian)
29. MKL: https://www.intel.com/content/www/us/en/docs/onemkl/developer-reference-c/2024-0/cblas-trsm.html#GUID-230D5D74-DB1B-469A-A24E-3F46FC8E41C7 (accessed: 11.08.2025)
30. OpenMP Application Programming Interface Examples Version 5.2 – April 2022: https://www.openmp.org/wp-content/uploads/openmp-examples-5-2.pdf (accessed: 11.08.2025)
31. SKIT-5: http://icybcluster.org.ua/index.php?lang_id=2&menu_id=5 (accessed: 11.08.2025)

Received 11.08.2025

Volodymyr Sydoruk,
PhD, senior researcher,
V.M. Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine,
https://orcid.org/0000-0003-0210-6020
wolodymyr.sydoruk@gmail.com

Oleksii Chystiakov,
PhD, senior researcher,
V.M. Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine,
https://orcid.org/0000-0001-6456-2094
alexej.chystyakov@gmail.com

Peter Benner,
Managing Director,
Max Planck Institute for Dynamics of Complex Technical Systems (MPI DCTS), Magdeburg,
https://orcid.org/0000-0003-3362-4103
benner@mpimagdeburg.mpg.de

Olena Nikolaievska,
PhD, senior researcher,
V.M. Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine.
https://orcid.org/0000-0002-5145-0189
elena_nea@ukr.net

УДК 519.6

Володимир Сидорук [1], Олексій Чистяков [1], Пітер Беннер [2], Олена Ніколаєвська [1 *]

Розв’язування систем лінійних рівнянь з блочною «хмарочосною» структурою

[1] *Інститут кібернетики імені В.М. Глушкова НАН України, Київ*
[2] *Інститут динаміки складних технічних систем імені Макса Планка, Магдебург, Німеччина*
[*] *Листування: elena_nea@ukr.net*

У цій статті представлено ефективний паралельний алгоритм розв’язування систем лінійних алгебраїчних рівнянь (СЛАР) із використанням блочно-«хмарочосних» матриць на обчислювальних системах із розділюваною пам’яттю. Алгоритм оптимізовано для високої продуктивності та масштабованості шляхом ефективного розподілу навантаження між процесорними ядрами. Його було протестовано на сучасному обчислювальному кластері під час розв’язування прикладних задач моделювання міцності будівельних конструкцій. Аналіз продуктивності охоплює час виконання, коефіцієнт прискорення та вплив розміру блоків. Результати показують значне скорочення часу обчислень і стабільну масштабованість зі збільшенням кількості ядер, що підтверджує придатність методу для великомасштабних інженерних симуляцій у галузях механіки конструкцій, прикладної фізики та інших обчислювально інтенсивних напрямів.

Ключові слова: високопродуктивні обчислення, система лінійних алгебраїчних рівнянь, метод Холецького, розріджені матриці, блочні матриці-хмарочоси, паралельні алгоритми, системи зі спільною пам'яттю.