Numerical Eigenmode Analysis Software for Microwave Guiding Structures

Dorogov Maksim
Dept. of Television and Control
Tomsk State University of Control Systems and Radioelectronics
Tomsk, Russian Federation
m.dorogof@gmail.com

Toropov Vladimir
Dept. of Television and Control
Tomsk State University of Control Systems and Radioelectronics
Tomsk, Russian Federation
vladimirtoropov516@gmail.com

Ivanov Anton
Dept. of Television and Control
Tomsk State University of Control Systems and Radioelectronics
Tomsk, Russian Federation
anton.a.ivanov@tusur.ru

***Abstract*—This paper presents novel software for the numerical eigenmode analysis of microwave guiding structures, developed in C++. The computational core of the software is based on a well-established two-dimensional Finite-Difference Frequency-Domain method. This approach formulates the eigenvalue problem, which is then solved using the Eigen and Spectra libraries. The software features a modular architecture based on object-oriented programming principles. The algorithm and graphical user interface are described in detail, including the main computational steps and key interface components. The software was tested on models of a microstrip transmission line, a rectangular waveguide and a broadside-coupled stripline. The obtained results were compared with those from commercial electromagnetic simulation software. The results showed excellent convergence, confirming the accuracy of the software. Future plans include expanding the software functionality, improving its computational efficiency, and performing robust comprehensive testing of its algorithms, including the simulation of various typical transmission lines and error handling during user interaction.**

***Keywords*—eigenmode, guiding structure, finite-difference frequency-domain method, C++, Eigen, Spectra.**

I. Introduction

Solving eigenvalue problems is an essential part of developing complex high-frequency systems. This is often done using modular computer-aided design (CAD) systems, which include tools for working with a graphical environment and numerical iterative algorithms for solving systems of linear algebraic equations (SLAEs). The importance of solving such problems is highlighted in several studies [1], [2], [3], [4], [5], [6], [7]. Unlike methods operating in the time domain, these studies employ the finite-difference method in the frequency domain to allow for the calculation of propagation constants as eigenvalues. To reduce memory consumption and computation time, this approach utilizes sparse matrices. Furthermore, current trends toward ensuring technological sovereignty require domestic solutions in electromagnetic modeling, which in turn highlights the shortage of such solutions in this area.

The purpose of this paper is to present software developed for the numerical eigenmode analysis of microwave guiding structures.

II. Software Development

A. Computational Core of the Software

The eigenmode analysis in the developed software is performed using the Finite-Difference Frequency-Domain method [8]. The method is based on representing Maxwell's equations in a two-dimensional finite-difference matrix formulation, which, after appropriate transformations and accounting for boundary conditions, reduces to a generalized eigenvalue problem

$$[\mathbf{A}]\cdot\{\mathbf{x}\} = \lambda[\mathbf{B}]\cdot\{\mathbf{x}\}, \qquad (1)$$

where [**A**] is a sparse coefficient matrix, [**B**] is the identity matrix, $\{\mathbf{x}\}=\{E_x, E_y, E_z, H_x, H_y, H_z\}^T$, $\lambda=k_0$ for the solution corresponding to the propagation constant, and $\lambda=\beta/k_0$ for the solution corresponding to cutoff frequencies.

The Eigen [9] and Spectra [10] libraries were selected to solve the SLAE (1). The Eigen library is a high-performance, template-based linear algebra library for C++ that supports dense and sparse matrices, including methods for computing eigenvalues. However, its efficiency proved insufficient for problems involving large sparse matrices, and therefore Spectra was used for these computations. This library builds on the functionality of Eigen and ARPACK [11] and is designed for solving large-scale eigenvalue problems.

B. Software Architecture

The software is based on object-oriented programming (OOP) and includes a set of interrelated classes. The *MainView* class acts as the main controller, managing interface tabs and user input. The *MainViewScene* class is responsible for the graphical display of the scene, interaction with geometric objects, and storage of initial calculation data (*MainData*). Geometric figures are created using the *CreateFigureRectangle*, *CreateFigureCircle*, and *CreateFigurePolygon* classes, each of which creates a figure of the corresponding type. The base class for all geometric figures is *ShapeItem*. Materials are managed via the *MaterialEdit* class, and the material library is controlled via the *MaterialLibraryDialog* class. Boundary conditions are modified using the *BordersEdit* class, the frequency range is set using the *FreqEdit* class, and calculation domain parameters are configured using the *SettingsEdit* class. Solver parameter configuration is performed by the *PortModeWindow* and *EigenModeWindow* classes, depending on the selected calculation type. The *SolverWorker* class performs calculations in a separate thread, which ensures

The research was carried out at the expense of Russian Science Foundation grant 23-79-10165, https://rscf.ru/project/23-79-10165/.

interface responsiveness during resource-intensive operations. The calculation results are passed to the *Results* class, where they are processed and presented in tabular form, while their graphical visualization is implemented using the *FieldViewWindow* class.

C. Software Algorithm

Fig. 1 shows the flowchart of the software algorithm. It illustrates the sequence of user actions: from specifying the calculation domain parameters, constructing the geometry, and assigning materials, to performing the calculation and subsequently visualizing the results.

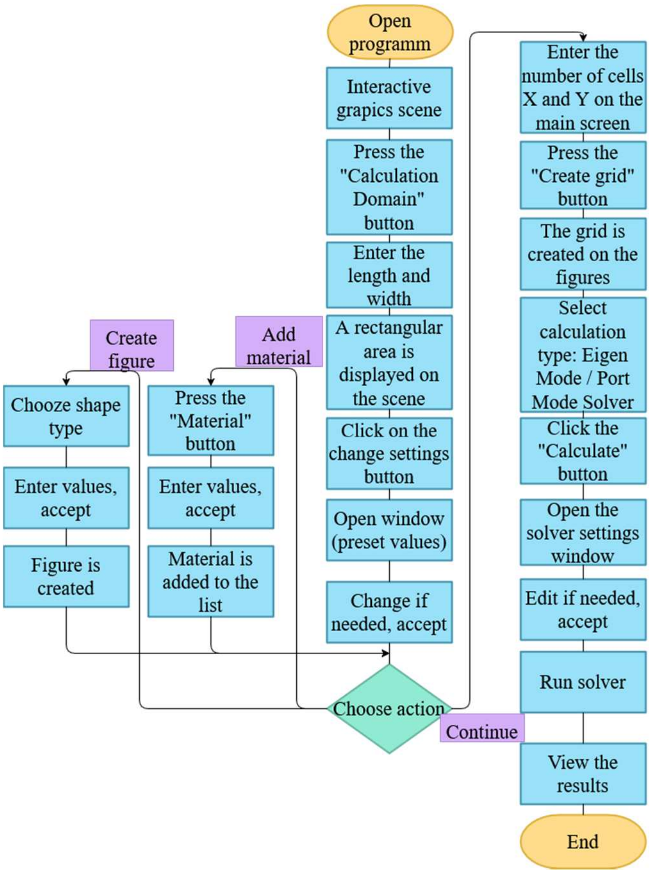

Fig. 1. Flowchart of the software's algorithm.

First, the user must specify the geometric dimensions of the rectangular computational domain, which define the spatial boundaries of the simulation. Next, the user must enter the frequency range and select boundary conditions or use the default values. After that, the user can create geometric shapes (rectangles, circles, and polygons) within the specified domain. For each shape, the user can assign a material, either creating a new one or selecting it from the material library. The computational mesh is generated based on the number of cells along the x and y axes specified by the user. The algorithm automatically calculates the actual dimensions of each cell, taking into account the boundaries of the geometric objects, ensuring optimal placement of mesh nodes relative to the boundaries of the shapes. Each cell is assigned a material index corresponding to the shape it lies within, thereby creating a material map. Next, the solver type is selected and its parameters are configured. The calculation is performed in a separate thread, and the results are displayed in tabular (cutoff frequencies or propagation constants) and graphical formats (the distribution of magnetic and electric fields).

D. Software User Interface

The graphical user interface and the software application logic were developed using the Qt framework. The user interface consists of a window divided into four main areas: the toolbar (Fig. 2), the graphics scene (Fig. 3), the shape list, and the material list (Fig. 4).

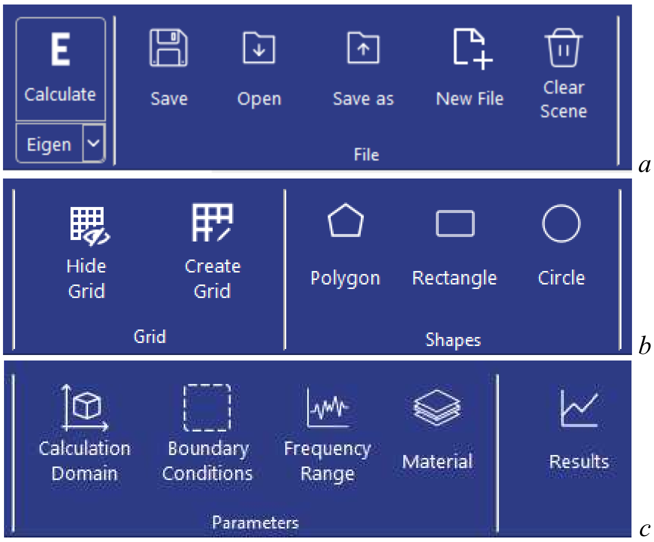

Fig. 2. Toolbar elements: for working with project files (*a*), for creating shapes and grids (*b*), for modifying settings and displaying results (*c*).

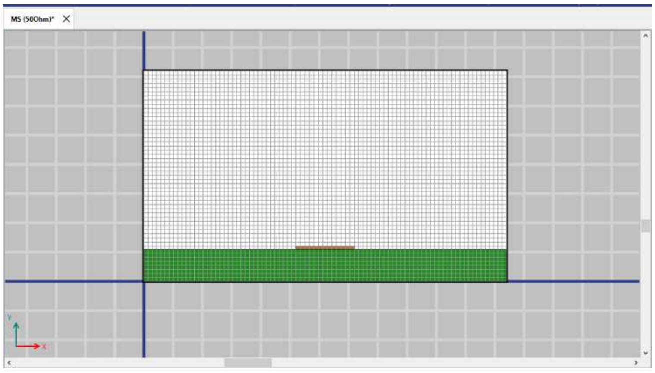

Fig. 3. The graphics scene in the software's main workspace.

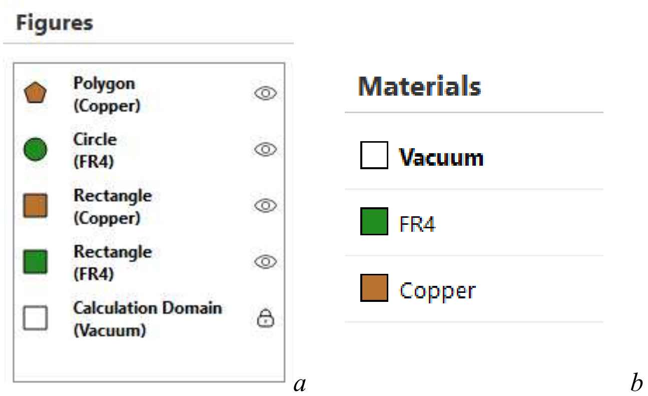

Fig. 4. List of figures (*a*) and list of materials (*b*)

The toolbar contains elements grouped by functionality: for working with project files, managing the grid, constructing geometric primitives, configuring parameters, performing calculations, and switching between the results tabs and the graphics scene. In the shape list, elements are displayed in sequence and are named according to their shape type and assigned material. Below the shape list are fields for entering grid parameters. The material list is generated in the order they are added, which determines their indexing when constructing the material map.

III. Software Testing

The software was tested on three models of guiding structures: a microstrip line (MSL), a rectangular hollow waveguide (HWG), and a broadside-coupled stripline (BCSL). The geometric parameters of the MSL are: a=12.4 mm, b=7.2 mm, t=0.1 mm, w=2 mm, h=1.1 mm, and for the BCSL, these are shown in Table I. Copper with an electrical conductivity of 58.1 MS/m was used as the conductor material. The cross-section of the HWG is defined by sides a=19.05 mm and b=9.525 mm. Schematic representations of the test models are shown in Fig. 5–7. For the HWG, cutoff frequencies were calculated in the range from 5 GHz to 20 GHz (Table II). For the MSL and BCSL, propagation constants were calculated at 25 GHz and 20 GHz, respectively (Table III and Table IV).

TABLE I. Geometric Parameters of the BCSL, mm

a	b	w	u	t	s	h	k
14	8	3	2.2	1	5	2	3

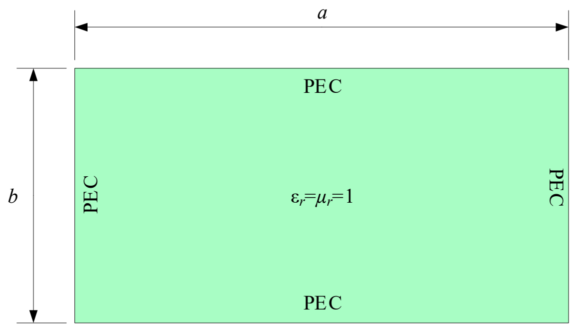

Fig. 5. HWG test model.

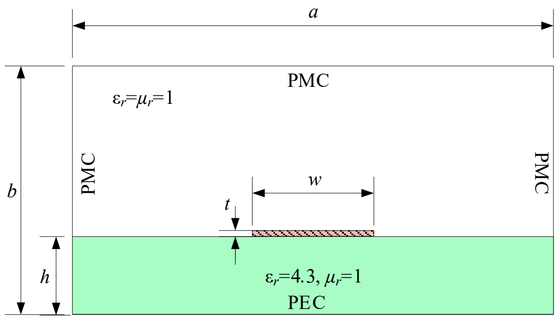

Fig. 6. MPL test model.

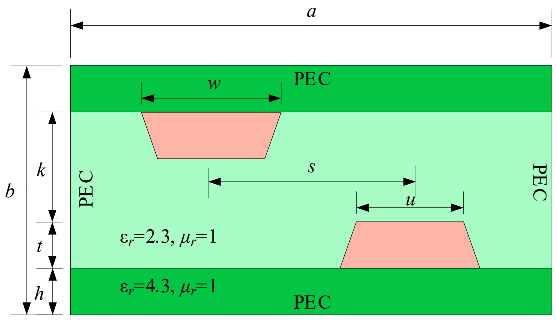

Fig. 7. BCSL test model.

The results obtained were compared with results from commercial electromagnetic simulation software (CEMSS) as well as with analytical calculations. The normalized distributions of the electric and magnetic fields for the test models are shown in Fig. 8–12. The results for the HWG are normalized to the maximum value, while those for the MSL and BCSL are normalized to 1 W of transmitted power [12].

TABLE II. Comparison of the Obtained Cutoff Frequencies, GHz

Mode	Our	CEMSS	Analytics
1	7.868	7.868	7.869
2	15.733	15.733	15.737
3	15.733	15.733	15.737
4	17.591	17.591	17.595
5	17.591	17.591	17.595

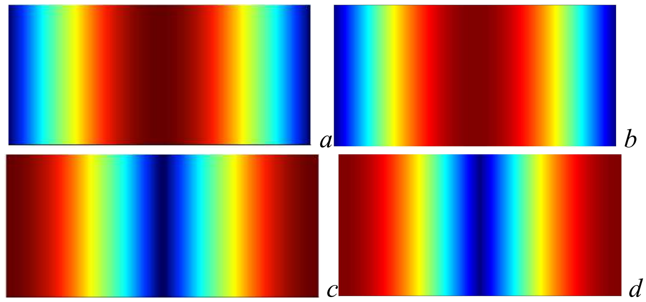

Fig. 8. Field distribution in HWG for the first mode: E_y Our (*a*), CEMSS (*b*); H_z Our (*c*), CEMSS (*d*).

TABLE III. Comparison of the Obtained MSL Propagation Constants, 1/m

Mode	Our	CEMSS
1	*j*1010.93	*j*1010.147
2	*j*594.584	*j*591.079
3	*j*556.356	*j*556.752
4	*j*437.069	*j*435.011
5	*j*307.701	*j*307.369

TABLE IV. Comparison of the Obtained BCSL Propagation Constants, 1/m

Mode	Our	CEMSS
1	827.578	827.583
2	683.618	683.619
3	638.71	638.688
4	622.279	622.251
5	585.734	585.712

IV. Conclusion

This work presents the developed software for the numerical eigenmode analysis of microwave guiding structures. Descriptions of the software computational core, architecture, operational algorithm, and user interface are provided. The software was tested on the MSL, HWG, and BCSL models. Cutoff frequencies were obtained as eigenvalues for the HWG, while propagation constants were calculated for the MSL and BCSL. Additionally, normalized spatial electromagnetic field distributions were calculated for all test models. A comparison of the results with results from CEMSS and the analytical solution showed excellent convergence, confirming the accuracy of the software. Future work will focus on expanding the functionality of the developed software, improving its computational efficiency, and performing comprehensive testing of its algorithms, including the simulation of various typical transmission lines and error handling during user interaction.

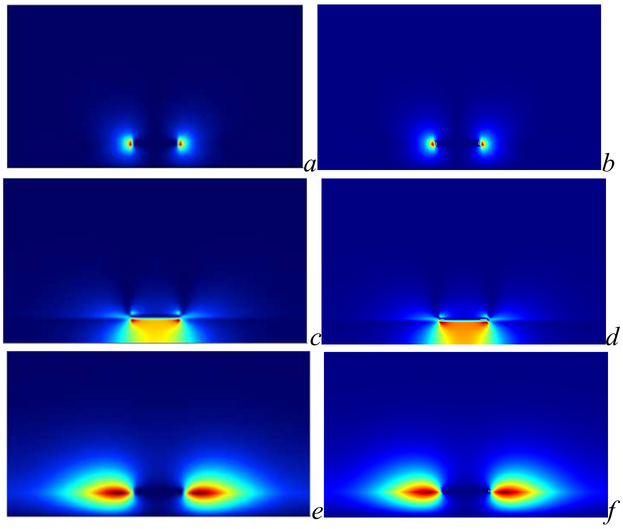

Fig. 9. Electric field distribution in the MPL for the first mode: E_x Our (*a*), CEMSS (*b*); E_y Our (*c*), CEMSS (*d*); E_z Our (*e*), CEMSS (*f*).

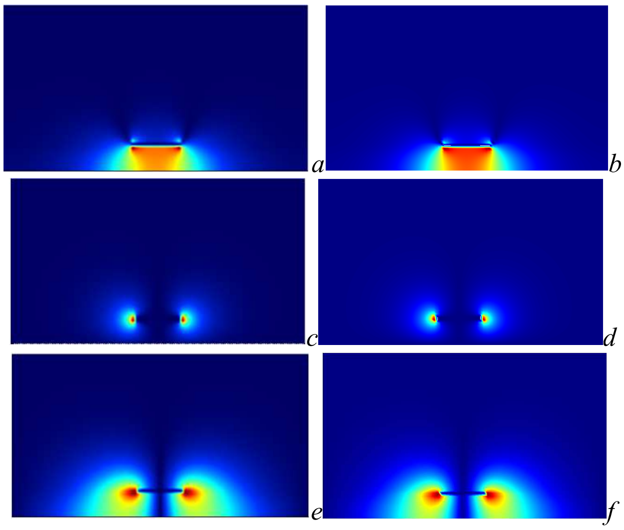

Fig. 10. Magnetic field distribution in the MPL for the first mode: H_x Our (*a*), CEMSS (*b*) H_y Our (*c*), CEMSS (*d*); H_z Our (*e*), CEMSS (*f*).

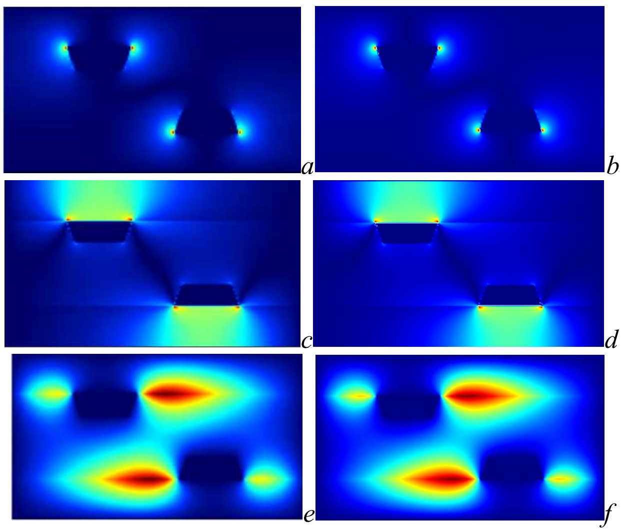

Fig. 11. Electric field distribution in the BCSL for the first mode: E_x Our (*a*), CEMSS (*b*); E_y Our (*c*), CEMSS (*d*); E_z Our (*e*), CEMSS (*f*).

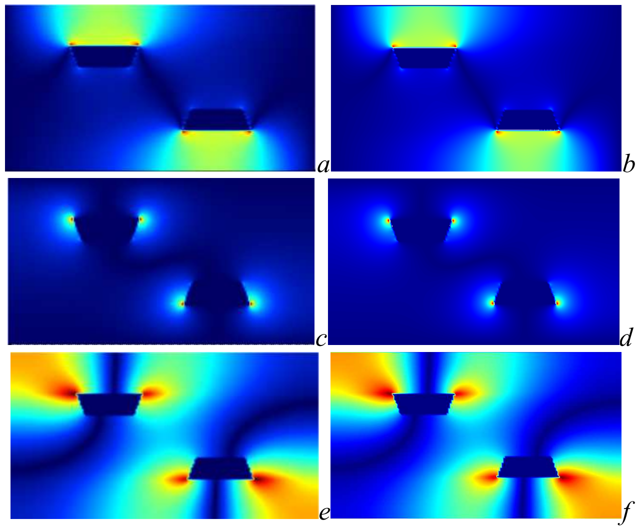

Fig. 12. Magnetic field distribution in the BCSL for the first mode: H_x Our (*a*), CEMSS (*b*) H_y Our (*c*), CEMSS (*d*); H_z Our (*e*), CEMSS (*f*).

REFERENCES

[1] G. Lin, T. Huang and X. Jin, "A Compact 2-D Conformal FDFD Method with Non-uniform Mesh for Waveguide Structures," 2022 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (AP-S/URSI), USA, 2022, pp. 938–939, doi: 10.1109/AP-S/USNC-URSI47032.2022.9887146.

[2] K. Wang, S. Zuo, Q. Wu, Z. Lin, Y. Zhang and X. Zhao, "A Novel Compact Conformal 2-D FDFD Method for Modeling Waveports in 3-D FDTD," in IEEE Antennas and Wireless Propagation Letters, vol. 23, no. 7, pp. 2091–2095, July 2024, doi: 10.1109/LAWP.2024.3381969.

[3] Y. Wang and S. Langdon, "A 2D FDFD eigenmode method with edge singularity correction and conformal technique," 2017 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting, USA, 2017, pp. 1339–1340, doi: 10.1109/APUSNCURSINRSM.2017.8072712.

[4] X. Gu, X. Jin, J. Li and B. Li, "Compact 2-D FDFD Analysis of Waveguide Structures with ARPACK," 2018 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting, USA, 2018, pp. 2379–2380, doi: 10.1109/APUSNCURSINRSM.2018.8608482.

[5] M. Liu, Z. Chen. "A direct computation of propagation constant using compact 2-D full-wave eigen-based finite-difference frequency-domain technique", Proceedings of the International Computational Electromagnetics Applications Conference, Beijing, 1999, pp. 78–81, doi: 10.1109/ICCEA.1999.825072.

[6] G. Lin, T. Huang, X. Jin [et al.]. "An Efficient Scheme of Waveguide Port Modeling Based on Conformal and Nonuniform Mesh for Time-Domain Finite Integration Technique", IEEE Transactions on Antennas and Propagation, 2025, vol. 73, no. 2, pp. 1047–1058, doi: 10.1109/TAP.2024.3511919.

[7] Y. Wang, S. Langdon. "A Two-Component 2-D FDFD Eigenmode Method Incorporated with the Conformal Technique", 2024 18th European Conference on Antennas and Propagation (EuCAP), 2024, pp. 1–4, doi: 10.1109/APUSNCURSINRSM.2017.8072712.

[8] Y. Zhao, K. Wu, and K. Cheng, "A compact 2-D full-wave finite-difference frequency-domain method for general guided wave structures," IEEE Transactions on Microwave Theory and Techniques, vol. 50, no. 7, pp. 1844–1849, July 2002, doi: 10.1109/TMTT.2002.800447.

[9] Eigen C++ Library. [Online]. Available: https://eigen.tuxfamily.org/index.php?title=Main_Page.

[10] Spectra – C++ Library For Large Scale Eigenvalue Problems. [Online]. Available: https://spectralib.org/quick-start.

[11] R. B. Lehoucq, D. C. Sorensen, C. Yang ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods. – Philadelphia: SIAM, 1998, 142 p, ISBN 978-0-89871-407-4.

[12] R. E. Collin, Foundations for Microwave Engineering, 2nd ed. NY, USA: Wiley-IEEE Press, 2001.