UDC 004.94
DOI: 10.15587/1729-4061.2025.322460

THE DEVELOPMENT OF AN ELECTRONIC CIRCUIT SIMULATION SYSTEM USING VARIABLE TABULAR BASES

Kostyantyn Kharchenko
PhD*
Oleksandr Beznosyk
Corresponding author
PhD*
E-mail: beznosyk.oleksandr@lll.kpi.ua
Yaroslav Kornachevskyi
PhD*
Bogdan Bulakh
PhD*
Vadym Yaremenko
Assistant*
*Department of System Design
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"
Beresteiskyi ave., 37, Kyiv, Ukraine, 03056

The object of the study is a system for electronic circuit simulation using variable tabular bases, which optimizes the calculation processes in complex circuits. The work addressed the problem of creating models of electronic circuits with a convenient description of components in a linear system of equations.

A method for electronic circuit simulation using variable tabular bases is presented. The problem addressed is improving the efficiency of simulation of electronic circuits of varying complexity through the use of two approaches: the General Tabular Model (GTM) and the Hybrid Model 10 (HM10).

The results obtained demonstrate that the GTM provides a universal approach to modeling due to the rapid formulation of equations but leads to an increase in the matrix dimensions. In contrast, the HM10 significantly reduces the number of variables in the equations, although it requires additional computations to achieve complete results. The system architecture includes the ability to change the tabular basis or create a custom tabular basis to adapt to specific electronic circuit modeling tasks, thanks to the efficient use of Python libraries such as NumPy.

The features and distinctive characteristics of the proposed system are the variability of tabular bases for optimal problem-solving, as well as the modularity of the system, which ensures its flexibility and scalability. Additionally, the system allows adding new components and adapting algorithms to specific scenarios.

The practical application of the results covers educational processes, scientific research, as well as automated analysis and optimization of the design of complex electronic devices. The system is effective for working with complex circuits containing nonlinear components and provides fast solutions to algebraic equations. The obtained results showed that for a certain circuit configuration, the size of the model matrix can be 1.5 to 2 times smaller, which significantly reduces the number of calculations

Keywords: electronic circuit simulation, circuit design, tabular bases, mathematical modelling, Python, NumPy

Received 22.11.2024
Received in revised form 13.01.2025
Accepted 30.01.2025
Published 24.02.2025

How to Cite: *Kharchenko, K., Beznosyk, O., Kornachevskyi, Y., Bulakh, B., Yaremenko, V. (2025). The development of an electronic circuit simulation system using variable tabular bases. Eastern-European Journal of Enterprise Technologies, 1 (4 (133)), 30–40. https://doi.org/10.15587/1729-4061.2025.322460*

1. Introduction

Modeling electronic circuits is an important component of the design process for modern electronic devices. As the complexity of these circuits continues to grow, traditional modeling methods face a number of challenges, particularly those related to computational time and calculation accuracy. In particular, classical methods used to solve linear equation systems may become inefficient when processing large electronic systems containing hundreds or even thousands of components [1, 2].

Most known electronic circuit simulation systems, such as SPICE, operate with only a single tabular model to form conductance matrices and systems of equations. This standardizes the modeling approach and ensures the stability of results across a wide range of tasks. However, using only one model has its limitations, especially when modeling complex or specialized electronic circuits. As noted in [3], a single-model approach may be ineffective for tasks where time optimization or the use of different approaches to equation system processing is critical.

For example, in complex circuits with a large number of elements and nodes, standard models may generate excessively large matrices, significantly slowing down the computational process. This creates additional difficulties in scaling systems, forcing researchers to seek alternative methods that balance modeling accuracy and performance [2].

One promising solution to this problem is the use of variable tabular bases, which enable more efficient equation system formation for modeling. The application of such models not only reduces the computational load but also enhances system flexibility when working with various types of electronic components.

Thus, integrating variable tabular bases into electronic circuit simulation systems should be explored to improve modeling flexibility and optimize computational resources. This is particularly relevant when dealing with large circuits, where classical methods lose their effectiveness.

Modern technologies also provide opportunities to use existing Python libraries, such as NumPy, to accelerate the simulation of complex circuits. This significantly reduces computation time and ensures high modeling accuracy, which is critically important in the context of electronic system design. Conversely, performing mathematical computations directly in Python without using libraries significantly slows down the circuit simulation process.

Research in the field of electronic circuit simulation is of great importance in today's world, as increasing device complexity necessitates the advancement of analysis and design methods. The introduction of variable tabular bases in modeling systems enables the resolution of tasks that traditional methods do not always handle efficiently, especially when working with large and complex systems. The practical significance of these results lies in accelerating computations, reducing equation system dimensionality, and providing flexibility in integrating various components. This opens up new possibilities for the development of innovative electronic devices, optimizing their operation, and reducing design costs. Such a modeling system can be used in educational process as well as for researching tabular models.

The relevance of this research topic lies in the need to improve tools for electronic circuit modeling, which are critically important for modern engineering [3, 4]. The increasing complexity of electronic devices, along with the integration of new materials and technologies, presents significant challenges for traditional analysis methods [5, 6]. Additionally, the relevance of using variable tabular bases in electronic circuit modeling is driven by the need to enhance the efficiency and flexibility of component descriptions within analytical systems.

2. Literature review and problem statement

In modern scientific literature, various approaches to constructing models of electronic circuits are considered, allowing for the specifics of different components to be taken into account.

In [2], a method of "growing" the corresponding linear system of equations for modeling an electronic circuit is studied, which makes it possible to construct an optimal matrix size for calculating voltage and current parameters on the components. However, this method forms the system of equations in a single way, which does not allow for the selection of a tabular model. This significantly limits the practical application of this approach.

The work [3] is devoted to new methods of modeling analog electrical circuits in the time domain, focusing on methods that efficiently use existing data structures for backward differentiation methods (BDF). The authors analyze modifications for hierarchical data structures that allow for accelerating the simulation of large circuits. Special attention is given to methods that improve the stability and accuracy of solving differential-algebraic equations, including the Radau, Rosenbrock-Wanner, and MEBDF methods. In conclusion, the authors outline the challenges associated with the increasing size and complexity of electronic systems and emphasize the importance of combining hierarchical and flat modeling approaches. However, this work is aimed at solving a narrow sector of problems related to time-domain modeling.

In [7], the modeling and analysis of electrical circuits are presented, including examples of series, parallel, and arbitrarily connected circuits. The input-output relationships of the main circuit elements – resistor, inductor, and capacitor – are presented in both the time and frequency domains. Kirchhoff's voltage and current laws are used in the analysis. To facilitate the analysis, the differential equation model of the circuits is transformed into a Laplace transform model. However, the work focuses on the physical modeling of systems, which is a unique feature and facilitates the understanding of system behavior. The possibility of using different tabular bases is not considered.

In [8], models of equivalent circuits are studied. In works on power and reliability analysis of electronic circuits, equivalent circuits are used to simplify calculations. These models are often used to solve problems in power systems and to model parameters such as currents and voltages in different parts of an electronic system. The research includes improvements to the calculation methods of such circuits to ensure higher accuracy of modeling results. However, the work is specifically dedicated to the modeling of power systems and does not consider other types of electronic circuits.

In projects [4, 9], an electrical circuit simulation system in the Python programming language is presented. However, the formation of the conductance matrix in such systems occurs in a single way, which can be a significant obstacle for studying complex systems where it is important to examine the corresponding linear equation matrices in different bases.

In [10], a modern approach to electrical circuit simulation is presented. The main goal of the project is to develop a tool that allows for easy and flexible modeling of nonlinear transistor circuits under transient conditions. The tool, named ReSPICE, provides efficient work with modern electronic components through accurate calculations of time dependencies and nonlinear effects. The project's code is open and available on the GitLab platform, which promotes the involvement of developers and researchers in its improvement. The project also offers a modern interface for integration into work environments, making it suitable for educational and research tasks, as well as for engineering use in the industry. The ability to change the formation of the linear system of equations in the system is not provided.

The work [11] examines the development of low-cost real-time hardware simulators designed for testing and improving power electronics control systems. The author proposes both analog and digital simulators, including those based on the TMS320F28388D microcontroller with a clock frequency of 200 MHz. These simulators allow for pre-testing control algorithms, reducing the risk of damaging expensive equipment during development. Special attention is given to the simulation of a three-phase shunt active filter (APF) as an example of a power electronics circuit. The proposed solutions demonstrate effectiveness in accelerating the development process and improving the reliability of power electronics control systems. However, the problem is considered in the context of power system control rather than general-purpose electronic circuits.

Regarding the educational component of such systems, the authors of [12] describe methods for use by students and researchers to study the principles of electronic systems operation. However, this work does not consider the practical application of the system in terms of performing calculations, for example, using Python or similar approaches.

Thus, modern research offers both traditional methods for solving systems of equations for electronic circuits and novel approaches. However, the main drawback of the aforementioned systems is the lack of an option to choose the method for forming the system of algebraic-differential equations since modern electronic circuit modeling tools offer the formation of a model in a single tabular basis that cannot be changed. In certain cases, this complicates the description of electronic circuit components and the inclusion of their elements in the matrix.

3. The aim and objectives of the study

The aim of this study is to develop a system for modeling electronic circuits with a flexible equation formulation system, specifically using the General Tabular Model (GTM) and Hybrid Model 10 (HM10) [5, 13], which allows for effective work with large data sets, ensuring speed and accuracy of simulation. This will enable the creation of an open-source Python toolkit for flexible modeling electronic circuits with a variable tabular basis (GTM and HM10), efficient analysis of complex systems, and integration into educational and scientific processes, ensuring versatility and optimization of computational resources.

To achieve this aim, the following objectives were set:

– develop the architecture of the electronic circuit simulation system with a variable tabular basis;

– describe the basic components of an electronic circuit based on algebraic-differential equations in different tabular bases;

– verify the correctness of the system's operation with a series of examples;

– investigate the advantages and disadvantages of using the General Tabular Model (GTM) and Hybrid Model 10 (HM10) for constructing the system of equations.

4. Materials and methods of research

This study investigates a system for modeling electronic circuits that uses variable tabular bases to form models and solve systems of linear equations describing electronic circuits of varying complexity. The main hypothesis of the research is that the use of variable tabular bases for modeling electronic circuits allows for more flexible description of circuit components and reduces the dimensionality of matrices.

During the research, the following assumptions were made:

– the circuits being modeled are linear, meaning their parameters do not depend on currents and voltages on other components (there are no dependent current and voltage sources);

– the components of the electronic circuits are described using linear models, where the relationships between voltage and current are linear;

– the circuits may contain differential components (capacitors and inductors).

For the convenience of the research, the following simplifications were adopted:

– nonlinear components (such as diodes or transistors) are not included in the base implementation of the system, but their addition is possible in future versions;

– thermal effects and other physical parameters of components are not considered;

– boundary conditions for complex circuits are simplified to the use of ideal voltage and current sources.

The Python programming language, known for its simplicity and readability, was chosen for the implementation of the modeling system allowing developers to quickly create and maintain code. This is especially important when developing complex systems, where code clarity and comprehensibility are critical. Python has a rich set of libraries for scientific computations and modeling, such as NumPy and TensorFlow. These libraries are optimized for numerical calculations and linear algebra, which significantly accelerates the process of electronic circuit simulation. While Python itself is sometimes considered relatively slow compared to compiled languages like C++, using libraries like NumPy and TensorFlow allows for mathematical computations to be performed at much higher speeds. This is achieved because these libraries are implemented in low-level programming languages and optimized for handling large amounts of data, as well as utilizing Python's mathematical libraries to speed up calculations. In this system, this means that mathematical computations (such as solving linear systems of equations) can be performed very quickly. Python supports object-oriented programming, enabling the creation of modular and extensible programs. This makes it easy to add new components and functionality to the simulation system without needing to rewrite the existing code. Python has a large and active developer community that provides continuous support and development of libraries and frameworks. This also makes it easier to find and implement the latest methods and approaches in the field of electronic circuit simulation. Additionally, Python is a cross-platform language, allowing created systems to run on different operating systems without significant changes to the code. This simplifies the development and use of the simulation system on various devices and platforms.

Several Python libraries were involved in the creation of the simulation system. The foundation for the mathematical computations in the system is the NumPy library. This library is the standard for scientific computing in Python. It provides efficient handling of multidimensional arrays and contains a wide range of functions for linear algebra, statistics, and other numerical methods. Using NumPy allows for fast and efficient execution of mathematical operations, such as solving linear systems of equations, which is one of the key tasks in numerical methods.

The *numpy.linalg.solve()* function provides an efficient way to solve systems of linear equations of the form $Ax=B$, where A is a square matrix of coefficients and B is a column vector of the right-hand sides of the equations. This function implements the LU decomposition method, which allows solving systems with minimal errors if the matrix is nonsingular. Using NumPy is the standard approach in scientific computing due to its simplicity and speed for small systems.

Since the main feature of the proposed simulation system is the ability to use two tabular models to form the system of equations describing the electrical circuit, let's briefly review the mathematical basis for forming these models.

The General Tabular Model (GTM) involves using the complete set of circuit variables, which includes the potential vector V, the current vector i, and the voltage vector u for all branches (a complete coordinate basis). Topological equations, based on Kirchhoff's first and second laws, are presented in matrix form using a structural matrix. These are supplemented by component equations reflecting the physical characteristics of the circuit elements. For numerical analysis, these equations are converted into linear algebraic form, after which each can be represented in one of two forms [7]:

– Y-component (current source):

$$i_j = \sum_k y_{jk} u_k + I_j; \tag{1}$$

– Z-component (voltage source):

$$u_j = \sum_k z_{jk} i_k + E_j. \tag{2}$$

In matrix form:

$$\begin{matrix} Branches \\ Nodes \\ Z \\ Y \end{matrix} \begin{pmatrix} A^T & 0 & -1 \\ 0 & A & 0 \\ 0 & -R_j & 1 \\ 0 & 1 & -G_j \end{pmatrix} \cdot \begin{pmatrix} V \\ i \\ u \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ E_j \\ I_j \end{pmatrix}. \tag{3}$$

Dimensionality of the system of equations:

$$(2n+m-1)\times(2n+m-1), \qquad (4)$$

where m is the number of nodes, n is the number of branches of the graph.

Values of currents and voltages:

$$i_j = G_j u_j + I_j, \qquad (5)$$

$$u_j = R_j i_j + E_j. \qquad (6)$$

The table-node model in potentials (hybrid model, HM10), also known as the extended homogeneous coordinate basis (EHCB), is the most widely used form for representing mathematical models of electrical circuits. It is used in many well-known circuit design packages, such as SPICE (PSpice, HSpice, T-Spice) and the CAD system ALLTED developed in Ukraine (NetAllted, WebAllted, GridAllted). Unlike HM1, where the basis variables are voltages across the edges of the graph, in HM10, node potentials are used. The number of potentials in the model is equal to n–1, where n is the number of nodes in the circuit, excluding the reference (most often zero) node. Thus, HM10 includes the vector of node potentials V and the vector of currents for Z-components i_z. The system of equations of the model consists of cross-section equations (taking into account the Y-component equations) and component equations for Z-components, and its dimensionality is n–1+m_z, where m_z is the number of Z-components [7].The structural matrix A, which reflects the connections between the nodes of the circuit (excluding the reference node), has the number of rows corresponding to the number of node potentials in the model. The voltages across all branches of the circuit can be easily expressed through the node potentials using the elements of matrix A:

$$u_v = V_i - V_j, \qquad (7)$$

where i and j are the indices of the positive and negative non-zero elements of the column of the structural matrix corresponding to branch V. If there is only one non-zero element in the column of the matrix, this means that the second node of the corresponding branch is the reference node, and the formula given above simplifies to a single component.

Accordingly, in matrix form, HM10 will have the following appearance:

$$\begin{pmatrix} AG_jA^T & A \\ -A^T & R_j \end{pmatrix} \cdot \begin{pmatrix} V \\ i \end{pmatrix} = \begin{pmatrix} I_j \\ E_j \end{pmatrix}. \qquad (8)$$

The difference between the formation of GTM and HM10 lies in the approaches to constructing and solving the systems of equations that describe electronic circuits. In GTM, the standard tabular approach is used to form the conductance matrix, which represents the connections between the elements of the circuit. This method is universal and suitable for a wide range of circuits, allowing for the quick formulation of equations for simulation. In GTM, the unknowns vector includes all currents and voltages across components, but the description matrix becomes quite large and sparse. In contrast to GTM, HM10 is a hybrid approach, where the unknowns vector includes fewer variables, but the process of constructing the matrix becomes more complicated. As a result, in HM10, additional calculations are required to find all the currents and voltages across the components.

5. Research results of the developed simulation system

5. 1. Architecture of the electronic circuit simulation system with a variable tabular basis

The structure of the developed simulation package is shown in Fig. 1, and the corresponding UML class diagram is shown in Fig. 2.

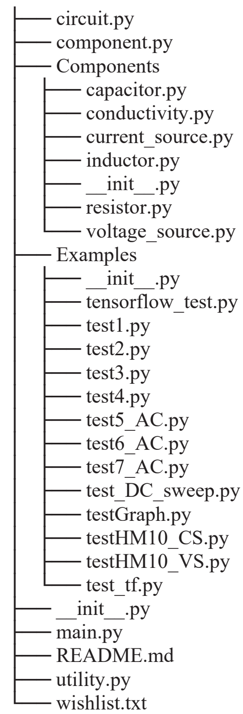

Fig. 1. Structure of the simulation system with a variable tabular basis

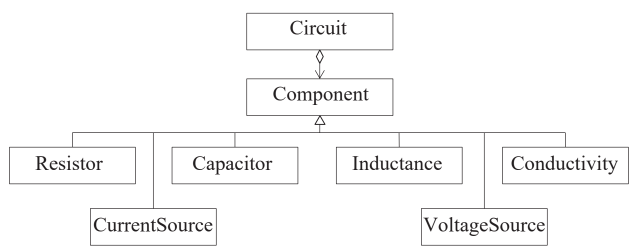

Fig. 2. UML class diagram of the simulation system with a variable tabular basis

The Circuit class (Fig. 3) is the main component of the electrical circuit simulation system developed in Python using scientific computing libraries. It implements basic functions such as adding components, calculating circuit nodes using the General Tabular Model (GTM) or Hybrid Model 10 (HM10), as well as solving linear systems of equations for DC and AC analysis. The class attributes, including arrays of matrices and vectors, ensure efficient data processing for mathematical modeling of complex electronic circuits. The use of the NumPy library in the class significantly enhances computational performance for large circuits. The primary feature of the class is its flexibility in modeling various components and its adaptability to user needs through a modular structure.

The Component class (Fig. 4) represents the basic element of an electrical circuit, which can be connected to the nodes of the circuit. It includes attributes for storing the element's identifier, a list of nodes it is connected to, a numerical parameter (e.g., resistance or current), and an index for unique identification.

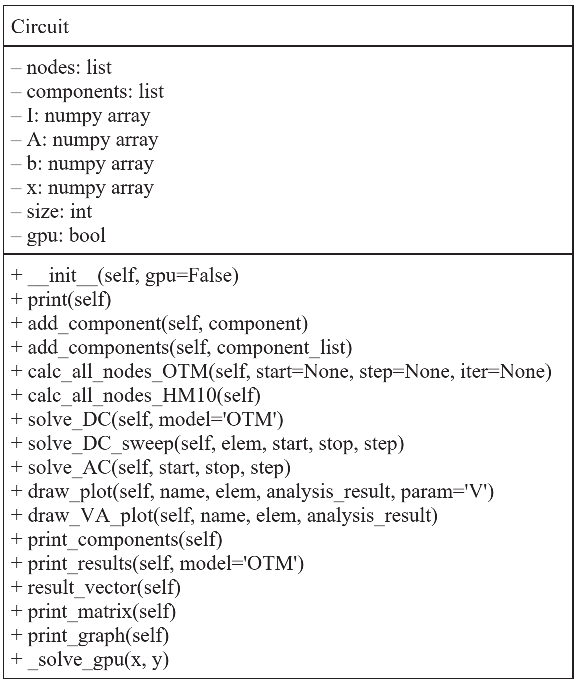

Fig. 3. UML class diagram of the Circuit class

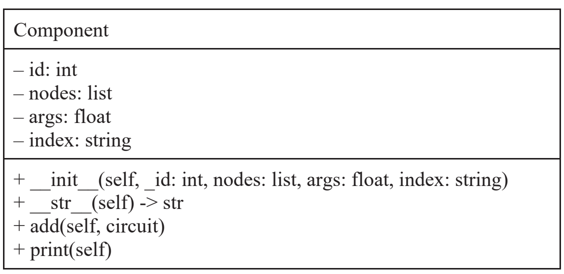

Fig. 4. UML class diagram of the Component class

The constructor of the class ensures the initialization of all these parameters. The *__str__* method allows representing the component as a textual description, making it easier to visualize. The *add* function integrates the component into the circuit, while *print* allows displaying information about it in a convenient format. The class serves as a base for modeling different types of components, such as resistors, capacitors, or power sources.

5. 2. Description of electronic circuit components in different tabular bases

The Resistor class (Fig. 5) models a resistor as one of the basic components of an electrical circuit. It inherits from the Component class, adding functions for integrating the resistor into the circuit using either the General Tabular Model (GTM) or the Hybrid Model 10 (HM10). The key parameters include the element's name, a list of nodes to which the resistor is connected, and a resistance value. The methods *add_GTM* and *add_HM10* implement the respective mathematical models for working with different types of circuits. The *print* function provides a textual representation of the resistor with all its characteristics.

The Conductivity class represents conductivity as a separate type of circuit element. This component also inherits from Component and has the main parameter of conductivity in siemens. The class methods are responsible for adding this element to the model and calculating its characteristics within the context of GTM and HM10. It is used for modeling components with an inverse dependence on resistance.

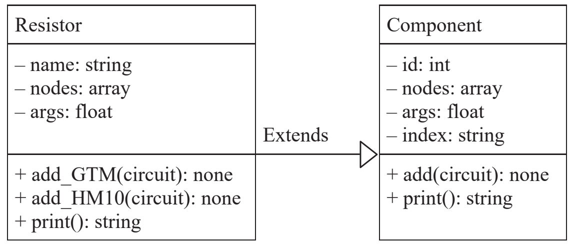

Fig. 5. UML diagram of the Resistor class

The Voltage Source class (Fig. 6) models a voltage source. The main attributes include the name, a list of nodes, a numerical voltage value, and a time-dependent (t) voltage function; possible functions:

$$V = V0 \cdot \sin\left(w \cdot t + \varphi\right), \tag{9}$$

$$V = \begin{cases} V0, & \text{for } t > t0, \\ 0, & \text{for } t \le t0. \end{cases} \tag{10}$$

The methods *add_GTM* and *add_HM10* integrate the voltage source into the circuit depending on the selected model. For alternating current analysis, support for time-dependent functions is implemented, allowing simulation of complex sources. The *time_func* method is used to calculate the instantaneous voltage values at a given moment in time.

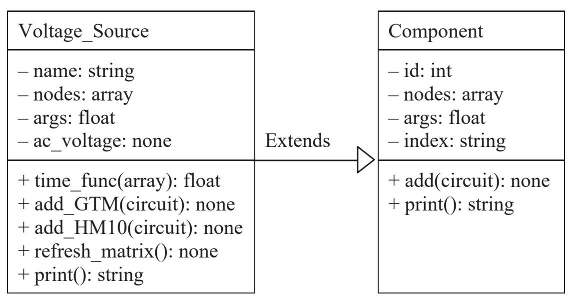

Fig. 6. UML diagram of the Voltage Source class

Similarly, the Current Source class (Fig. 7) is designed for modeling a current source. Like the voltage source, it includes mandatory parameters such as the name, nodes, a current value, as well as a time-dependent (t) function for simulation of alternating current; possible functions:

$$I = I0 \cdot \sin\left(w \cdot t + \varphi\right), \tag{11}$$

$$I = \begin{cases} I0, & \text{for } t > t0, \\ 0, & \text{for } t \le t0. \end{cases} \tag{12}$$

The implemented methods allow adding the current source to both the GTM and HM10 models. The *time_func* function is responsible for generating current values depending on time, enabling the simulation of dynamic processes.

The Capacitor class (Fig. 8) models a capacitor. Its main attributes include the name, nodes, a capacitance value, and an initial charge value. The *add_GTM* and *add_HM10* methods integrate the capacitor into the circuit, taking into account the capacitance equations. For time-domain analyses, support for computations based on Euler's method is implemented. The class allows simulation of the charging and discharging dynamics of the capacitor.

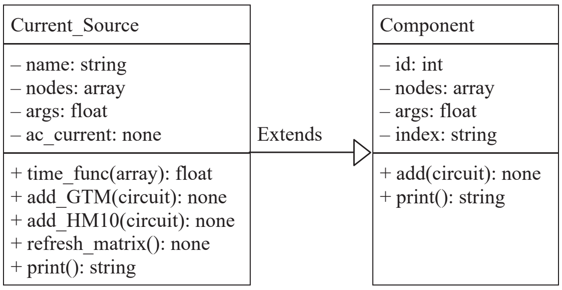

Fig. 7. UML class diagram of the Voltage Source

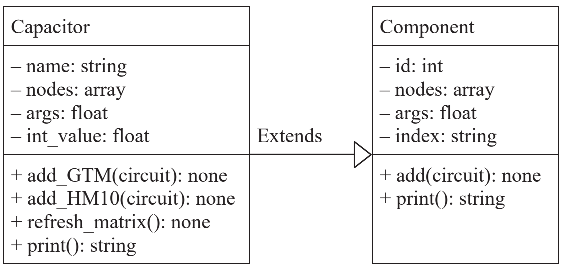

Fig. 8. UML diagram of the Capacitor class

The equation for linear capacitance is as follows:

$$I(t)=Cu'(t). \tag{13}$$

Thus:

$$I(t)=C\frac{\partial u}{\partial t}. \tag{14}$$

To perform time analysis with alternating current, it is necessary to replace the derivative value with an approximation. Using the explicit Euler method:

$$x_{n+1}=x_n+h\cdot f(x,t), \tag{15}$$

this can be rewritten as:

$$I=\frac{C}{\Delta t}u_{n+1}-\frac{C}{\Delta t}u_n. \tag{16}$$

And, accordingly, the Inductor class (Fig. 9) is responsible for modeling inductance. Its main parameters include the name, nodes, an inductance value, and an initial current. Like the Capacitor class, it supports the *add_GTM* and *add_HM10* methods for integration into different models. For alternating current analysis, it implements current calculation through inductance using numerical methods, such as the explicit Euler method. This allows the investigation of transient processes in circuits with inductance.

The equation for linear inductance is as follows:

$$V(t)=Li'(t). \tag{17}$$

Thus:

$$U(t)=L\frac{\partial i}{\partial t}. \tag{18}$$

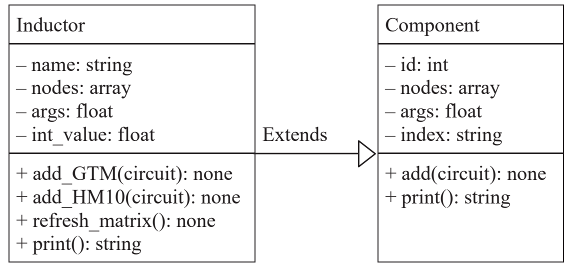

Fig. 9. UML diagram of the Inductor class

Similarly, to the case of capacitance, for time analysis with alternating current, it is necessary to replace the derivative value with an approximation. Using the explicit Euler method:

$$x_{n+1}=x_n+h\cdot f(x,t), \tag{19}$$

this can be rewritten as:

$$U=\frac{L}{\Delta t}i_{n+1}-\frac{L}{\Delta t}i_n. \tag{20}$$

5. 3. Examples of electrical circuits simulation

In this work, several examples of electrical circuit simulation were considered, by forming both GTM and HM10, and the obtained results were compared.

Example 1. The circuit (Fig. 10) consists of two elements: a current source and a resistor, and two nodes (one of which is zero).

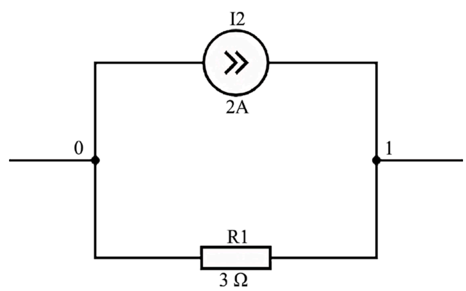

Fig. 10. Circuit for Example 1

Code for describing the circuit configuration:

```
R1=Resistor('R1', [0, 1], 3) # 3 Ohm
I2=CurrentSource('I2', [0, 1], 2) # 2 Ampere
test_circuit=Circuit()
test_circuit.add_components([R1, I2])
```

Let the current of the source be 2 A and the resistance of the resistor be 3 Ω. Let's conduct a DC analysis for these values:

```
test_circuit.solve_DC() # DC
```

The system of equations formed using the GTM method looks as follows:

$$\begin{pmatrix} -1 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & -1 \\ 0 & -1 & -1 & 0 & 0 \\ 0 & -R1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} V1 \\ i_{R1} \\ i_{I2} \\ u_{R1} \\ u_{I2} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ I2 \end{pmatrix}. \tag{21}$$

The system of equations formed using the HM10 method looks as follows (the matrix is 1×1):

$$\left(\frac{1}{R1}\right)\cdot\left(V1\right)=\left(I2\right). \quad (22)$$

Results for Example 1:

```
GTM
[[−1.  0.  0. −1.  0.]   x1   [0.]
 [−1.  0.  0.  0. −1.]   x2   [0.]
 [ 0. −1. −1.  0.  0.] * x3 = [0.]
 [ 0. −3.  0.  1.  0.]   x4   [0.]
 [ 0.  0.  1.  0.  0.]]  x5   [2.]
V1=6.0 Volt.
R1=3 Ohm.
i(R1)=−2.0 Ampere | u(R1)=−6.0 Volt.
I2=2 Ampere.
i(I2)=2.0 Ampere | u(I2)=−6.0 Volt.

HM10
[[−0.33333333]] * x1=[2.]
V1=−6.0 Volt.
R1=3 Ohm.
```

Example 2. The circuit (Fig. 11) consists of four components (a voltage source and three resistors) and three nodes (one of which is zero).

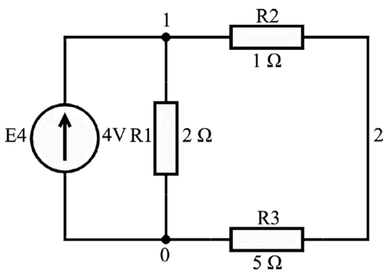

Fig. 11. Circuit for Example 2

Code for describing the circuit configuration:

```
R1=Resistor('R1', [1, 0], 2) # 2 Ohm
R2=Resistor('R2', [1, 2], 1) # 1 Ohm
R3=Resistor('R3', [2, 0], 5) # 5 Ohm
E4=VoltageSource('E4', [1, 0], 4) # 4 Volt
test_circuit=Circuit()
test_circuit.add_components([R1, R2, R3, E4])
```

Let the value of the voltage source be 4 V, and the resistances of resistors R1, R2, and R3 be 2 Ω, 1 Ω, and 5 Ω, respectively.

The hybrid method HM10 does not provide a complete basis of element values, so some values need to be calculated additionally. A DC analysis was conducted for these values:

```
test_circuit.solve_DC('HM10') # DC
```

The system of equations formed using the HM10 method is as follows:

$$\begin{pmatrix} \frac{1}{R1}+\frac{1}{R2} & -\frac{1}{R2} & 1 \\ -\frac{1}{R2} & \frac{1}{R2}+\frac{1}{R3} & 0 \\ 1 & 0 & 0 \end{pmatrix}\cdot\begin{pmatrix} V1 \\ V2 \\ i_{E4} \end{pmatrix}=\begin{pmatrix} 0 \\ 0 \\ E4 \end{pmatrix}. \quad (23)$$

At the same time, the corresponding GTM is expectedly much larger:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -R1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -R2 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -R3 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} V1 \\ V2 \\ i_{R1} \\ i_{R2} \\ i_{R3} \\ i_{E4} \\ u_{R1} \\ u_{R2} \\ u_{R3} \\ u_{E4} \end{pmatrix}=\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ E4 \end{pmatrix}. \quad (24)$$

Results of Example 2:

```
GTM
[[ 1.  0.  0.  0.  0.  0. −1.  0.  0.  0.]   x1   [0.]
 [ 1. −1.  0.  0.  0.  0.  0. −1.  0.  0.]   x2   [0.]
 [ 0.  1.  0.  0.  0.  0.  0.  0. −1.  0.]   x3   [0.]
 [ 1.  0.  0.  0.  0.  0.  0.  0.  0. −1.]   x4   [0.]
 [ 0.  0.  1.  1.  0.  1.  0.  0.  0.  0.]   x5   [0.]
 [ 0.  0.  0. −1.  1.  0.  0.  0.  0.  0.] * x6=[0.]
 [ 0.  0. −2.  0.  0.  0.  1.  0.  0.  0.]   x7   [0.]
 [ 0.  0.  0. −1.  0.  0.  0.  1.  0.  0.]   x8   [0.]
 [ 0.  0.  0.  0. −5.  0.  0.  0.  1.  0.]   x9   [0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  1.]]  x10  [4.]
V1=4.0 Volt.
V2=3.333333333333 Volt.
R1=2 Ohm.
i(R1)=2.0 Ampere | u(R1)=4.0 Volt.
R2=1 Ohm.
i(R2)=0.666666666667 Ampere | u(R2)=0.666666666667 Volt.
R3=5 Ohm.
i(R3)=0.666666666667 Ampere | u(R3)=3.333333333333 Volt.
E4=4 Volt.
i(E4)=−2.666666666667 Ampere | u(E4)=4.0 Volt.

HM10
[[ 1.5 −1.   1. ]   x1   [0.]
 [−1.   1.2  0. ] * x2=[0.]
 [ 1.   0.   0. ]]  x3   [4.]
V1=4.0 Volt.
V2=3.333333333333 Volt.
R1=2 Ohm.
```

Example 3. The circuit (Fig. 12) consists of four components (a current source, a resistor, an inductor, and a capacitor) and three nodes (one of which is zero).

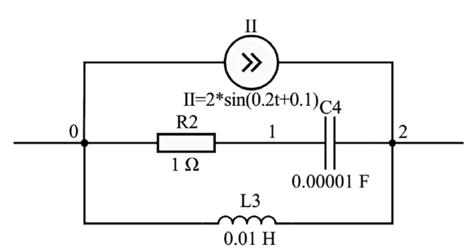

Fig. 12. Circuit for Example 3

The code for describing the circuit configuration:

```
I1=CurrentSource('I1', [2, 0], 2, (2, 0.2, 0.1)) # 2 Ampere
R2=Resistor('R2', [1, 0], 1) # 1 Ohm
L3=Inductor('L3', [2, 0], 0.01) # 0.01 Henry
C4=Capacitor('C4', [1, 2], 0.00001) # 0.00001 Farad
test_circuit=Circuit()
test_circuit.add_components([I1, R2, L3, C4])
```

Let the value of the source be given by the formula (11), where $I0=2$, $w=0.2$, $\varphi=0.1$.

The resistance of the resistor is 1 Ω, the inductance is 0.01 H, and the capacitance is 10 μF. Let's perform an AC analysis for these values:

```
res=test_circuit.solve_AC(0, 0.001, 0.000002)
```

The time for the analysis was set as follows: the start time is 0 s, the end time is 0.001 s, and the time step is 0.000002 s. The system of equations formed using the GTM method for the first iteration is as follows:

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -R2 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{L3}{\Delta t} & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & \frac{C4}{\Delta t} \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ i_{I1} \\ i_{R2} \\ i_{L3} \\ i_{C4} \\ u_{I1} \\ u_{R2} \\ u_{L3} \\ u_{C4} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ I1 \\ 0 \\ \frac{L3 \cdot i_{L3}^{n-1}}{\Delta t} \\ \frac{C4 \cdot u_{C4}^{n-1}}{\Delta t} \end{pmatrix}. \quad (25)$$

The HM10 model is accordingly given as follows:

$$\begin{pmatrix} \frac{1}{R2}+\frac{C4}{\Delta t} & -\frac{C4}{\Delta t} & 0 & 0 \\ -\frac{C4}{\Delta t} & \frac{C4}{\Delta t} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & \frac{L3}{\Delta t} \end{pmatrix} \cdot \begin{pmatrix} V1 \\ V2 \\ i_{I1} \\ i_{L3} \end{pmatrix} = \begin{pmatrix} \frac{C4 \cdot u_{C4}^{n-1}}{\Delta t} \\ -\frac{C4 \cdot u_{C4}^{n-1}}{\Delta t} \\ I1 \\ \frac{L3 \cdot i_{L3}^{n-1}}{\Delta t} \end{pmatrix}. \quad (26)$$

The simulation results are shown in Fig. 13.

The results of the electrical circuit simulation demonstrate the time-dependent variations of current and voltage across different elements of the circuit. In Fig. 13, *a*, the current through capacitor C4 is shown, characterized by harmonic oscillations in phase with the source's driving current. The graph in Fig. 13, *b* displays the voltage across inductor L3, where a typical sinusoidal dependence is observed with a minimum value at the midpoint of the period. The graph in Fig. 13, *c* shows the voltage across resistor R2, which increases according to a harmonic law. The results indicate the accurate reproduction of transient processes in the electrical circuit, accounting for all reactive elements.

Table 1 presents the characteristics of test electronic circuits in different tabular models.

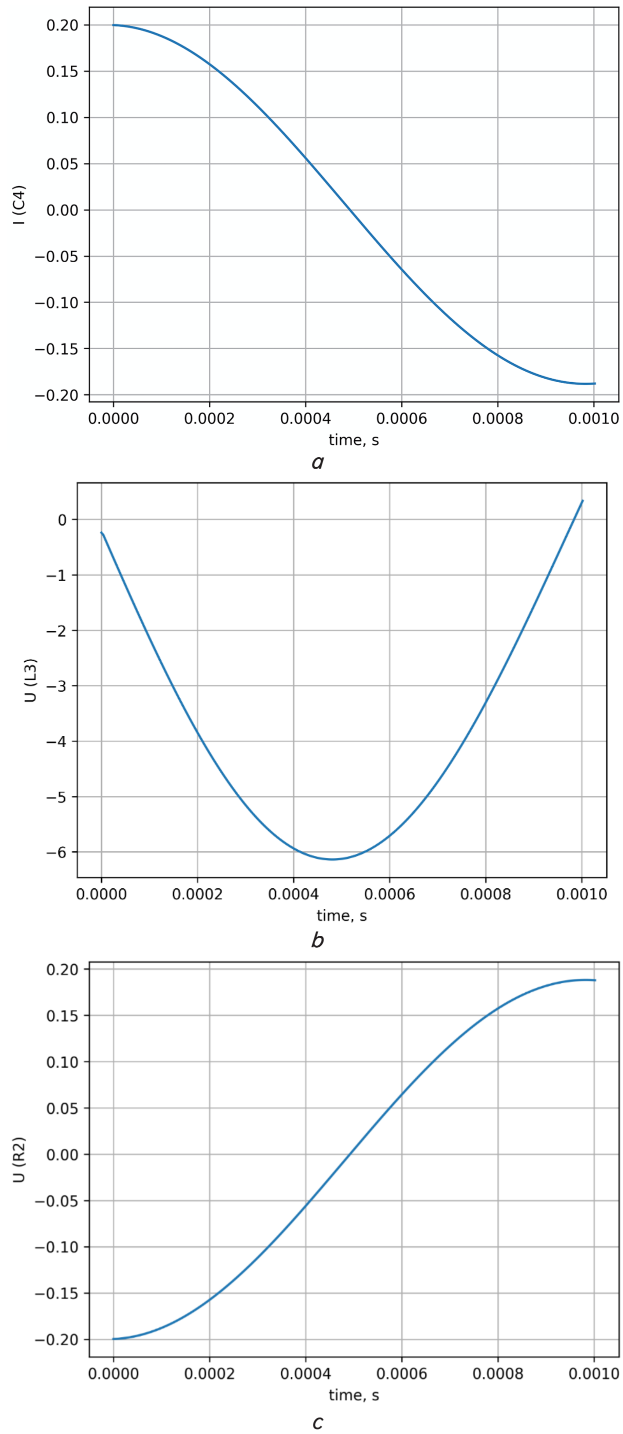

Fig. 13. Simulation results of the circuit with an alternating current source: *a* – current on capacitor C4; *b* – voltage on inductor L3; *c* – voltage on resistor R2

Table 1

The characteristics of test electronic circuits

Example	The number of components in the circuit	The number of nodes in the circuit	The number of equations in the GTM model	The number of equations in the HM10 model
1	2	1	5	1
2	4	3	10	3
3	4	3	10	4

From the table, it is evident that the number of equations in the tabular GTM model significantly exceeds the corresponding number in the HM10 model. The GTM model provides a convenient way to describe *Z*- and *Y*-components without adapting to a specific circuit. On the other hand, in the HM10 tabular model, describing *Z*-components that depend on node voltages is more complex. Despite the larger size of the GTM tabular model, solving the linear system of equations immediately yields currents and voltages for all components. If there is a need to reduce the size of the linear system, the user can choose the HM10 tabular model or create their own by inheriting from the tabular model class in Python within the circuit.py file.

5. 4. Advantages and disadvantages of the key features of the proposed system

Advantages and disadvantages of the proposed simulation system with variable tabular bases are shown in Table 2.

Table 2

Advantages and disadvantages of building a simulation system with variable tabular bases

Feature	Advantages	Disadvantages
Use of the General Tabular Model	The General Tabular Model (GTM) provides a universal approach to modeling electronic circuits by using the conductance matrix. It is suitable for a wide range of circuits, allowing for the rapid formulation of equations for simulation	Large matrix size, leading to a high number of computations for the linear equation system
Use of the Hybrid Model 10	Hybrid Model 10 (HM10), unlike GTM, employs a smaller system of equations, reducing the number of variables but making calculations more complex. This requires additional computations to obtain complete simulation results	Complexity in describing certain components. Additional calculations are required for currents and voltages across components that are not included in the variable vector of the HM10 model
Use of the Numpy library for computations	Numpy enables efficient solutions to large linear equation systems without requiring additional hardware	Slower performance compared to GPU usage
Expansion capabilities	Python allows for easy expansion of the system by adding new components and functionality without rewriting the code, ensuring flexibility and modularity in development	Python's potential slowness in some cases requires careful implementation, ensuring that core computations are performed using NumPy or TensorFlow
Practical application	The developed system has been tested on various electronic circuit examples, confirming its ability to accurately model different electronic components such as resistors, capacitors, current sources and voltage sources	Requires basic Python programming experience to work with the system effectively

The table highlights the key challenge of balancing the universality of the approach and computational efficiency in electronic circuit simulation. The General Tabular Model (GTM) provides a universal approach for equation formulation across a wide range of circuits but suffers from a significant increase in matrix size, complicating computations for complex circuits. In contrast, the Hybrid Model 10 (HM10) reduces the number of variables and optimizes computations, but at the cost of additional steps required to obtain a complete set of results.

6. Discussion of the results of the research on the developed electronic circuit simulation system

The research results confirm the feasibility of developing a system for modeling electronic circuits using variable tabular bases. The proposed system (Fig. 1, 2) is quite flexible and easily scalable to account for new components and to create a custom tabular basis. The developed classes correspond to the most common components of electronic circuits, enabling the simulation of both linear and reactive elements. The presented examples demonstrate the correctness of the obtained results (including those shown in Fig. 13), as well as confirm the effectiveness of the chosen approaches.

The study demonstrated that the General Tabular Model (GTM) allows for the quick and universal formation of a system of equations for various circuits. However, it is characterized by large matrix dimensions, significantly increasing the computational resources required for solving them (Example 1, equations (21) and (22)). In contrast, the Hybrid Model 10 (HM10) demonstrates a significant reduction in the number of variables in the system of equations (Example 2, equations (23) and (24)). However, HM10 requires additional calculations to obtain complete results, especially when it is necessary to calculate currents and voltages across all components not included in the main variable vector.

The comparison of the General Tabular Model (GTM) and the Hybrid Model 10 (HM10) (Table 1) demonstrated their unique advantages and disadvantages depending on the complexity of the electronic circuits. GTM allows for quick equation formation due to the universal approach to modeling, but its use can lead to large matrices, affecting the speed of the solution.

On the other hand, HM10 reduces the dimensionality of the system of equations but requires additional calculations to obtain complete results, which can be less efficient when simulation of circuits with a large number of components.

Unlike existing systems for electronic circuit simulation, the key feature of the proposed system is the variability of tabular bases for optimal solution of various tasks, including the possibility of creating custom basis, as well as the modularity of the system, which ensures its flexibility and scalability by allowing the addition of new components and functions without the need to rewrite the main code. This is an important aspect for expanding the functionality and adapting the system to new requirements in future research and development.

In work [3], it was demonstrated that the use of a standard tabular model provides universality of approach but significantly increases computational costs when analyzing complex circuits. The obtained results confirmed these conclusions but also showed that the use of the Hybrid Model 10 allows for a significant reduction in the number of variables and improves computational efficiency. Additionally, the researches [2, 9] emphasized the importance of optimizing the matrix structure for complex electronic systems, which aligns with the proposed approach of changing tabular bases. Furthermore, the use of Python libraries such as NumPy was effective in reducing computation time, as noted in work [4]. Thus, the developed simulation system takes into account the best practices described in the literature, while offering a new approach to analyzing complex electronic circuits.

Thus, by using different tabular models within the developed system, the problem of modeling circuits of various sizes and quality compositions is solved, which will require the use of appropriate tabular bases for more convenient modeling (including from the user's perspective in terms of component description complexity).

Overall, the developed system objectively cannot compete with well-known commercial systems for electronic circuit simulation but has demonstrated high convenience for use and can be successfully applied both within scientific research and in educational process.

The limitation of the practical application of the proposed system is the lack of capabilities for modeling circuits with nonlinear components (such as controlled current and voltage sources, diodes, transistors, etc.), but the existing system architecture allows for the future addition of such components.

The disadvantage of the developed system is the need to describe the rules for adding each circuit component if the user wants to create a custom tabular basis.

Further research will focus on the following:

– investigating the calculation speed for electronic circuits with a large number of components;

– investigating the calculation speed using GPU for solving systems of linear equations;

– exploring the possibility of adding an AC and DC analysis module using nonlinear components.

7. Conclusions

1. In the course of the research, the architecture of the simulation system with different tabular bases was developed. The system was implemented using the Python programming language and relevant libraries. The main advantages of the system are its flexibility, scalability, and the ability to perform calculations using fast mathematical libraries. The system's architecture allows for the inclusion of new tabular bases and user-defined circuit components. The possibility of forming a variable tabular basis was demonstrated using two models (GTM and HM10). The proposed simulation system is not a competitor to systems like Spice, ALLTED, and others; it is specifically designed for scientific research and educational process.

2. Models of electronic circuit components in different tabular bases were described and implemented in corresponding classes in Python. In total, 6 main components for electronic circuit simulation were implemented (resistor, conductance, current and voltage sources, capacitor, inductance). The user has the ability to easily implement their own components and include them in the proposed types of analysis.

3. The correctness of the developed system operation was verified by simulating several electrical circuits in AC and DC analysis modes. The results obtained fully correspond to the theoretical calculations for building GTM and HM10, with the values of currents and voltages on the circuit components matching in both models. The simulation results are provided both numerically and graphically for AC analysis. Testing showed that, for certain classes of circuits, using HM10 allows reducing the dimensionality of the system of equations by 1.5–2 times compared to GTM, significantly reducing computational costs. However, the tabular model of GTM has its advantages, as it conveniently allows both *Y*-components and *Z*-components to be entered into the matrix.

4. The considered General Tabular Model (GTM) and Hybrid Model 10 (HM10) demonstrated different performance aspects. GTM, due to its universality, provides rapid equation formation for a wide range of electronic circuits, but it can lead to the formation of large matrices and decreased solution speed when simulating complex systems. HM10, on the other hand, uses smaller systems of equations, reducing the number of calculations, but requires additional calculations to obtain complete results, which may be less optimal for large circuits. The results confirmed the effectiveness of using variable tabular bases in modeling electronic circuits. Using two different models allowed the advantages and disadvantages to be identified depending on the circuit's nature and simulation requirements. Thus, the proposed approach provides flexibility in choosing tabular models and allows for adapting the modeling process to the user's specific needs.

Conflict of interest

The authors declare that they have no conflict of interest concerning this research, including financial, personal, authorship, or any other aspects that could potentially affect the research and its results presented in this article.

Financing

The research was conducted without financial support.

Data availability

The manuscript does not contain any associated data.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies in the creation of the work presented.

References

1. Petrenko, A. I., Ladogubets, V. V., Tchkalov, V. V., Pudlowski, Z. J. (1997). ALLTED: A computer-aided engineering system for electronic circuit design. Melbourne Vic Australia: UNESCO International Centre for Engineering Education (UICEE), 205.
2. Tarabarov, S. B. (1998). Modeliuvannia elektronnykh skhem metodom formuvannia kompaktnoi matrytsi providnostei. Visnyk NTUU "Kyivskyi politekhnichnyi instytut". Radiotekhnika, radioaparatobuduvannia, 32, 48–50. Available at: https://radap.kpi.ua/radiotehnika/article/view/968/923

3. Fijnvandraat, J. G., Houben, S. H. M. J., ter Maten, E. J. W., Peters, J. M. F. (2006). Time domain analog circuit simulation. Journal of Computational and Applied Mathematics, 185 (2), 441–459. https://doi.org/10.1016/j.cam.2005.03.021
4. Venturini, G. A SPICE-like electronic circuit simulator written in Python. GitHub. Available at: https://github.com/ahkab/ahkab
5. Kharchenko, K. V., Taranenko, A. V. (2022). Rozrobka dodatku dlia modeliuvannia ta analizu elektrychnykh kil. Zbirnyk dopovidei I naukovo-praktychnoi konferentsiyi "Systemni nauky ta informatyka", 377–381. Available at: http://mmsa.kpi.ua/sites/default/files/systemni_nauky_ta_informatyka_2022.pdf
6. Razavi, B. (2016). Design of Analog CMOS Integrated Circuits. McGraw-Hill: Education, 800. Available at: https://www.academia.edu/76121731/Design_of_Analog_CMOS_Integrated_Circuits_Behzad_Razavi_z_lib_org
7. Sundararajan, D. (2022). Mathematical Modeling of Electrical Systems. Control Systems, 45–61. https://doi.org/10.1007/978-3-030-98445-8_3
8. Jovicic, A., Jereminov, M., Pileggi, L., Hug, G. (2018). An Equivalent Circuit Formulation for Power System State Estimation including PMUs. 2018 North American Power Symposium (NAPS), 1–6. https://doi.org/10.1109/naps.2018.8600633
9. Iyer, S. V. (2018). Simulating Nonlinear Circuits with Python Power Electronics. Springer International Publishing. https://doi.org/10.1007/978-3-319-73984-7
10. Krüger, M. (2024). Flexible and easy to use non-linear transient electric circuit simulator. GitLab. Available at: https://gitlab.com/Makman2/respice/
11. Sozański, K. (2024). Low-Cost Hardware Analog and Digital Real-Time Circuit Simulators for Developing Power Electronics Control Circuits. Energies, 17 (24), 6359. https://doi.org/10.3390/en17246359
12. Salehizadeh, M., Nouri, H. (2020). Circuit Modelling by Difference Equation: Pedagogical Advantages and Perspectives. Mathematical Modelling of Engineering Problems, 7 (1), 26–30. https://doi.org/10.18280/mmep.070104
13. Kharchenko, K., Beznosyk, O., Kornachevskyi, Y., Taranenko, A. Electronic, mechanic and etc. modeller. GitHub. Available at: https://github.com/konst1970/dnipro