Original article
UDC 621.3.049
DOI: https://doi.org/10.18127/j19997493-202502-02

Performance and accuracy analysis of linear solvers in circuit-level simulation

V.S. Melikyan[1], **E.S. Ghazaryan**[2], **A.G. Harutyunyan**[3]

[1–3] National Polytechnic University of Armenia (Yerevan, Armenia)
[1] CJSC Synopsys Armenia (Yerevan, Armenia)
[1] vazgenm@synopsys.com, [2] eghazaryan911@gmail.com, [3] harash@seua.am

Abstract

Problem Statement. Accurate and efficient circuit modeling is crucial in modern electronics. A key part of circuit simulation is solving large systems of linear equations generated using Modified Nodal Analysis (MNA), where circuit behavior is represented through Kirchhoff's laws. Efficient and accurate numerical solvers are essential, especially for large-scale circuits with varying sparsity and condition numbers. The challenge lies in selecting an appropriate solver that balances computational efficiency, memory usage, and accuracy based on matrix properties such as symmetry, sparsity, condition number, and size.

Objective. The main goal of this work is a comparative analysis of four numerical solvers: LU decomposition, Sparse LU, Conjugate Gradient (CG), and an adaptive method combining CG and Sparse LU. The evaluation is based on performance and accuracy analysis using C++ Eigen library solvers, considering execution time, memory consumption, and numerical accuracy.

Results. Direct solvers (LU, Sparse LU) provide high accuracy but suffer from high computational and memory costs, making them impractical for large circuits. Sparse LU improves efficiency by leveraging matrix sparsity, making it a preferable choice when exact solutions are required. Iterative solvers, particularly CG, exhibit superior scalability for highly sparse systems, achieving the lowest execution times. The Adaptive method dynamically switches between CG and Sparse LU, offering a balance between efficiency and accuracy.

Practical Significance. The obtained results enable engineers to make correct decisions when selecting solvers based on their circuit characteristics, ensuring both computational efficiency and numerical reliability.

Keywords

Adaptive method, accuracy and performance, LU, Sparse LU, Conjugate Gradient (CG), circuit, computational resources and memory

The research was supported by the Higher Education and Science Committee of MESCS RA (Research project № 25RG-2B002).

For citation

Melikyan V.S., Ghazaryan E.S., Harutyunyan A.G. Performance and accuracy analysis of linear solvers in circuit-level simulation. Dynamics of complex systems. 2025. V. 19. № 2. P. 7–11. DOI: https://doi.org/10.18127/j19997493-202502-02 [in Russian]

Introduction

As is commonly known that accurate and efficient circuit simulation [1] is essential for circuit design validation, optimization and performance evaluation, which helps detect potential issues at early stages [2].

Once the circuit model is made, it must be converted into a mathematical representation to find solution to the key quantities such as node voltages, currents, and power dissipation. SPICE (Simulation Program with Integrated Circuit Emphasis) [3] plays a central role in this transformation, converting the circuit level representation into a system of equations that can be solved to analyze the circuit's behavior.

Modified Node Analysis (MNA) [4] is the most common method of converting circuit representations into mathematical equations . By applying Kirchhoff's Current Law (KCL) and Kirchhoff's Voltage Law (KVL) circuit equations are derived and formulated as a system of linear equations expressed in the classical $Ax = B$ matrix form for linear equation solving.

The article focuses on how these equations are formulated, solved, and optimized using LU decomposition, Sparse LU [6], Iterative Conjugate Gradient (CG) [2], and an Adaptive solver [2,5,6] (that combines CG and Sparse LU), with particular focus on DC analysis. It also shows the effect of matrix size, sparsity, symmetry, and condition number on the process's performance and accuracy. By understanding these factors and employing appropriate techniques, engineers can ensure that performed simulations are both accurate and efficient, leading to reliable and optimized circuits models.

Performance and Accuracy Analysis of Linear Equation Solvers

In this section, performance analysis of LU decomposition, Sparse LU [6], Iterative Conjugate Gradient (CG), and an Adaptive method for solving the system $Ax = B$ is implemented and as well as different strengths and weaknesses [7] depending on the characteristics of the matrix A are shown, including its size, sparsity, and condition number. C++ Eigen library which has an efficient implementation for matrix operations is used to analyze the performance and accuracy of the given linear equation solvers.

Performance comparison across matrix sizes

This section represents the dependence of each solver's running time on the matrix size. To ensure the parallelization [5] of matrix operations, experiments are conducted using 4 processor cores, which allow parallelization of both LU and Sparse LU factorization [5, 6, 8].

In order to reflect realistic problem scenarios, randomly generated symmetric matrices with 50% sparsity and an average condition number were used for the analysis. This analysis provided insights into how each solver scales and when it is most appropriate to use it based on its execution time.

Fig. 1 shows the dependence of execution time (in seconds) on the size of different matrices:

The results show that solver selection depends on matrix properties and computational constraints. For small matrices (≤500), LU decomposition or Sparse LU [6] is suitable for exact solutions. For large matrices, LU becomes computationally expensive and less efficient, reflecting its $O(n^3)$ complexity. For large, symmetric positive-definite matrices, CG is the most efficient due to its scalability and low computational cost. For large matrices, the Adaptive Method shows an average runtime that reflects the combined behavior of both solvers, depending on the matrix's convergence behavior.

In addition to execution time, memory usage is a critical factor when selecting a solver, particularly for large-scale simulations. Fig. 2 illustrates the CPU memory consumption (peak memories in KB) for each solver as the matrix size increases.

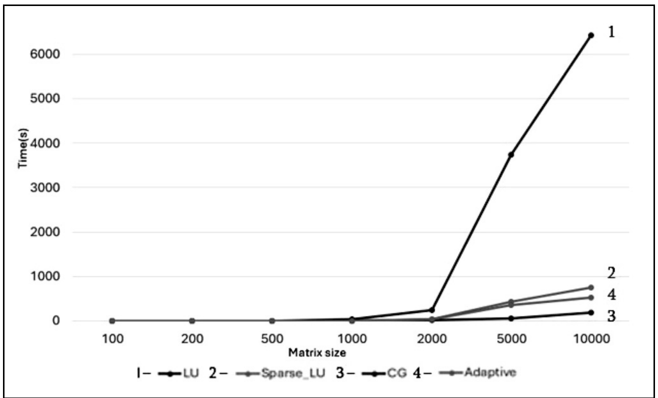

Fig. 1. Execution time vs. matrix size

Рис. 1. Графическое представление зависимости времени выполнения вычислений от размера матрицы

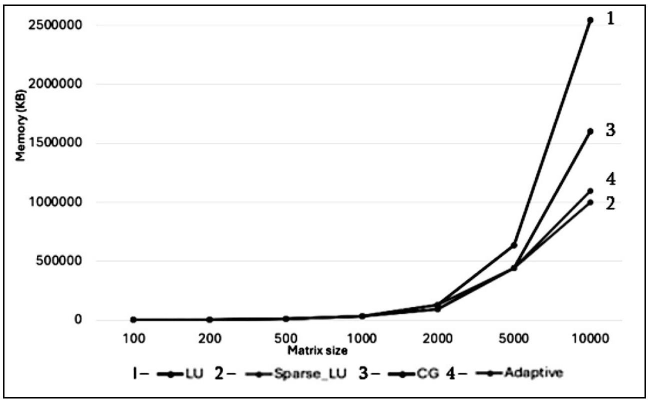

Fig. 2. Memory consumption across solvers for different matrix sizes

Рис. 2. Графическое представление зависимости потребления памяти разными решателями для разных размеров матриц

The analysis shows that LU decomposition exhibits the highest memory consumption due to full matrix factorization. In contrast, Sparse LU reduces memory usage by leveraging sparsity but still requires more memory than iterative solvers. The CG method stands out with the lowest memory footprint, as it only stores a few vectors. The Adaptive Method begins with a low memory profile similar to CG, but when switching to Sparse LU, memory usage temporarily increases, though it manages overall memory efficiently by releasing matrix A after factorization.

Analysis of execution time based on matrix sparsity

Fig. 3 represents how by keeping the matrix size fixed (1000×1000) and maintaining a medium condition number the sparsity level of the matrix affects the execution time of different solvers.

The results demonstrate that direct solvers such as LU and Sparse LU are suitable for denser matrices, while iterative solvers like CG and Adaptive are more efficient in case sparsity increases [5,6]. For highly sparse matrices, the CG and Adaptive methods emerge as the most efficient choices due to their minimal execution time.

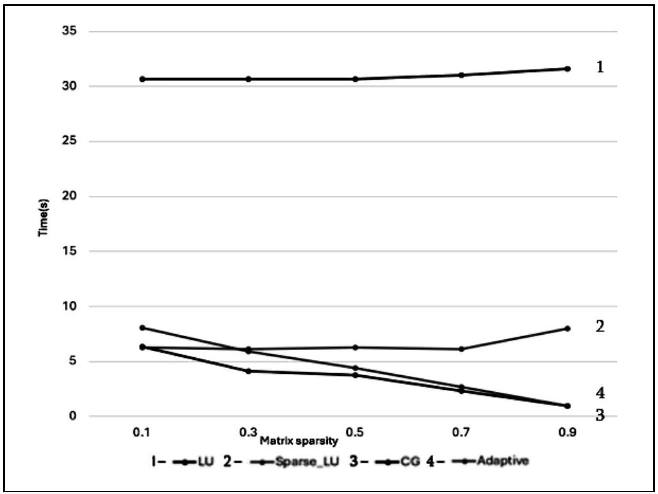

Fig. 3. Execution time vs. matrix sparsity

Рис. 3. Графическое представление зависимости времени выполнения от разреженности матрицы

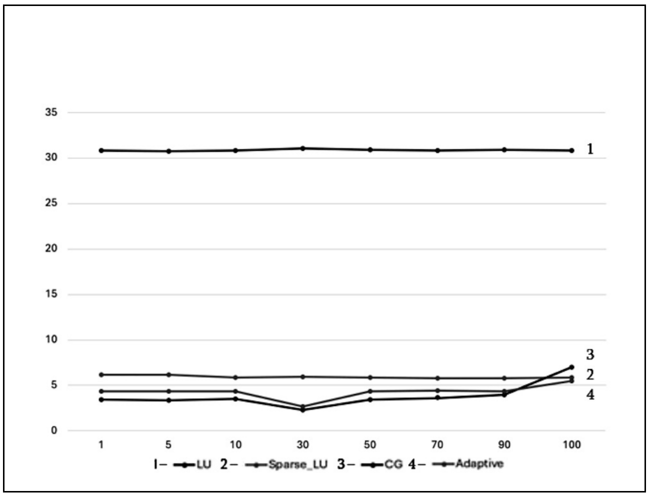

Fig. 4. Execution time vs. condition number

Рис. 4. Графическое представление зависимости времени выполнения от номера условия

Execution time dependence on matrix condition number

Fig. 4 explores how the matrix condition number affects solver performance while keeping the matrix size fixed at 1000×1000 with 50% sparsity.As the condition number increases, iterative methods require more iterations, making them slower. In contrast, direct solvers maintain a constant running time, which makes them more reliable for ill-conditioned problems.

Accuracy comparison of direct and iterative methods

To evaluate the accuracy of different solvers, used LU decomposition as the reference method, considering its ability to produce exact solutions without approximation errors. The accuracy of other solvers is assessed by computing the relative solution error with respect to the LU solution:

$$\text{Relative Error} = \frac{\left\| x_{\text{solver}} - x_{\text{reff}} \right\|}{\left\| x_{\text{reff}} \right\|}.$$

Since symmetric matrices are commonly used in circuit-level simulation, 1000×1000 symmetric matrix with varying sparsity levels are generated while using a moderate condition number, to investigate the impact of sparsity on accuracy. For each generated matrix, the system is solved by using different solvers and computing the relative error between their solutions and the reference LU solution.

Table 1 presents the accuracy results of different solvers by measuring the relative solution error compared to the reference LU decomposition method.

Table 1. ***Relative solution error across different solvers***

Sparsity levels	0,1%	0,3%	0,5%	0,7%	0,9%
Sparse LU	$1{,}66269 \cdot 10^{-13}$	$1{,}72294 \cdot 10^{-13}$	$1{,}56812 \cdot 10^{-13}$	$1{,}11647 \cdot 10^{-13}$	$1{,}38168 \cdot 10^{-13}$
CG	$1{,}0687 \cdot 10^{-4}$	$1{,}10202 \cdot 10^{-5}$	$1{,}6532 \cdot 10^{-7}$	$1{,}00013 \cdot 10^{-8}$	$1{,}085 \cdot 10^{-9}$
Adaptive	$1{,}456 \cdot 10^{-3}$	$2{,}1533 \cdot 10^{-5}$	$1{,}005 \cdot 10^{-6}$	$1{,}6643 \cdot 10^{-7}$	$1{,}7324 \cdot 10^{-8}$

The results highlight that direct solvers (LU, Sparse LU) provide the most accurate solutions, while iterative solvers (CG, Adaptive) are less accurate in their computational efficiency [7]. The accuracy of CG and Adaptive solvers improves with increasing sparsity, making them more suitable for highly sparse matrices. The Adaptive method provides a balance between efficiency and robustness, although its accuracy remains slightly lower than CG in some cases.

Conclusion

The analysis of linear equation solvers for circuit-level simulation emphasizes key aspects of computational efficiency, memory usage, and accuracy. The performance analysis showed that while the LU decomposition is robust, it is computationally expensive for large circuits. Sparse LU improves efficiency when matrix sparsity level increases, making it a preferable choice for structured circuit matrices. Conjugate Gradient and Adaptive methods achieved the lowest execution times due to their excellent scalability for highly sparse systems.

The memory usage analysis for LU decomposition demonstrated the highest memory, limiting its adaptation for large systems, whereas Sparse LU reduces storage requirements but still exceeds those of iterative methods. Based on results CG is most memory-efficient solver, while the Adaptive method initially follows CG's low memory profile but temporarily increases storage when switching to Sparse LU, balancing efficiency with stability.

Direct solvers, such as LU and Sparse LU, maintain high accuracy, making them suitable for applications where precision is critical. However, CG showed increasing accuracy as matrix sparsity grows. The Adaptive method balances speed and accuracy, offering a versatile solution for engineers seeking both computational efficiency and numerical reliability in circuit simulations.

References

1. *Najm F.N.* Circuit simulation. John Wiley & Sons. 2010.
2. *Hamedi-Hagh S.* Computational Electronic Circuits Simulation and Analysis with MATLAB. 2022.
3. *Hewlett J.D., Wilamowski B.M.* SPICE as a Fast and Stable Tool for Simulating a Wide Range of Dynamic Systems. International Journal of Engineering Education. 2011. V. 27. № 2. P. 217–224.
4. *Hanke M.* An Introduction to the Modifed Nodal Analysis. 2008. August 20. P. 1–10.
5. *Chen X., Wang Y., Yang H.* A Fast Parallel Sparse Solver for SPICE-based Circuit Simulators. Design, Automation & Test in Europe Conference & Exhibition (DATE). 2015. P. 205–209.
6. *Chen X., Xia L., Wang Y., Yang H.* Sparsity-Oriented Sparse Solver Design for Circuit Simulation. Design, Automation & Test in Europe Conference & Exhibition (DATE). 2016. P. 1583–1584.
7. *Shastri K.D., Biswas R., Kumari P.* Comparison of Direct and Iterative Methods of Solving System of Linear Equations. JSRD - International Journal for Scientific Research & Development. 2015. V. 3. № 04. P. 3287–3288.
8. *Kapre N., DeHon A.* Parallelizing Sparse Matrix Solve for SPICE Circuit Simulation using FPGAs. IEEE International Conference on Field-Programmable Technology (FPT 2009), December 9-11.

Information about the authors

Vazgen S. Melikyan – Corresponding Member of National Academy of Science of RA, Senior Member IEEE, Head of Department, Dr. Sc. (Eng.), Professor
Elen S. Ghazaryan – Post-graduate Student, Software Engineer
Ashot G. Harutyunyan – Dr. Sc. (Eng.), Professor

The article was submitted 21.04.2025
Approved after reviewing 13.05.2025
Accepted for publication 26.05.2025

Научная статья
УДК 621.3.049
DOI: https://doi.org/10.18127/j19997493-202502-02

Анализ производительности и точности линейных решателей при моделировании схем

В.Ш. Меликян[1], **Э.С. Казарян**[2], **А.Г. Арутюнян**[3]

[1–3] Национальный политехнический университет Армении (г. Ереван, Республика Армения)
[1] ЗАО «Синопсис Армения» (г. Ереван, Республика Армения)
[1] vazgenm@synopsys.com, [2] eghazaryan911@gmail.com, [3] harash@seua.am

Аннотация

В современном моделировании схем точное и эффективное моделирование систем имеет большое значение. Одним из ключевых моментов моделирования является решение больших систем линейных уравнений, которые генерируются с помощью модифицированного узлового анализа (MNA), где поведение цепи моделируется законами Кирхгофа. Для решения этих уравнений необходимы эффективные и точные численные решатели, особенно для крупномасштабных схем с переменной разреженностью и числами обусловленности.

Проблема состоит в выборе подходящего метода, который сбалансирует вычислительную эффективность, использование памяти алгоритма и точность на основе таких свойств матрицы, как симметрия, разреженность(sparsity), число обусловленности (condition number) и размер матрицы. В работе рассмотрены прямые и итерационные численные методы: LU, Sparse LU, сопряженный градиент (CG), а также адаптивный метод, который представляет собой комбинацию CG и Sparse LU. Производительность этих решателей анализируется на основе времени выполнения алгоритма, используемой памяти и числовой точности.

Прямые решатели, основанные на результатах (LU, Sparse LU), обеспечивают высокую точность, но страдают как от высоких вычислительных затрат, так и от затрат памяти, что делает их непрактичными для больших схем. Sparse LU повышает эффективность за счет использования разреженности матрицы, что делает его предпочтительным выбором, когда требуются точные решения. Итерационные решатели, в частности CG, демонстрируют превосходную масштабируемость для очень разреженных систем, достигая кратчайшего времени выполнения. Адаптивный метод, объединяющий CG и Sparse LU и способный динамически переключаться между ними, предлагает компромисс между эффективностью и точностью.

Результаты показывают, что при выборе решателя следует руководствоваться характеристиками матрицы схемы. Для небольших матриц предпочтительны прямые решатели, такие как LU и Sparse LU, если точность является приоритетом, в то время как для больших, разреженных, симметричных положительно определенных матриц наиболее эффективным выбором являются CG и адаптивный метод, который обеспечивает баланс между скоростью и устойчивостью за счет динамической адаптации к свойствам сходимости матрицы.

Полученные результаты предоставляют инженерам системный подход к выбору решателей на основе структуры схемы, гарантируя как вычислительную эффективность, так и численную точность.

Ключевые слова

Адаптивный метод, точность и производительность, LU, Sparse LU, Conjugate Gradient (CG), схема, вычислительные ресурсы и память

Для цитирования

Меликян В.Ш., Казарян Э.С., Арутюнян А.Г. Анализ производительности и точности линейных решателей при моделировании схем. Динамика сложных систем. 2025. Т. 19. № 2. С. 7–11. DOI: https://doi.org/10.18127/j19997493-202502-02

Список источников

1. *Najm F.N.* Circuit simulation. John Wiley & Sons. 2010.
2. *Hamedi-Hagh S.* Computational Electronic Circuits Simulation and Analysis with MATLAB. 2022.
3. *Hewlett J.D., Wilamowski B.M.* SPICE as a Fast and Stable Tool for Simulating a Wide Range of Dynamic Systems. International Journal of Engineering Education. 2011. V. 27. № 2. P. 217–224.
4. *Hanke M.* An Introduction to the Modifed Nodal Analysis. 2008. August 20. P. 1–10.
5. *Chen X., Wang Y., Yang H.* A Fast Parallel Sparse Solver for SPICE-based Circuit Simulators. Design, Automation & Test in Europe Conference & Exhibition (DATE). 2015. P. 205–209.
6. *Chen X., Xia L., Wang Y., Yang H.* Sparsity-Oriented Sparse Solver Design for Circuit Simulation. Design, Automation & Test in Europe Conference & Exhibition (DATE). 2016. P. 1583–1584.
7. *Shastri K.D., Biswas R., Kumari P.* Comparison of Direct and Iterative Methods of Solving System of Linear Equations. JSRD – International Journal for Scientific Research & Development. 2015. V. 3. № 04. P. 3287–3288.
8. *Kapre N., DeHon A.* Parallelizing Sparse Matrix Solve for SPICE Circuit Simulation using FPGAs. IEEE International Conference on Field-Programmable Technology (FPT 2009), December 9-11.

Информация об авторах

Вазген Шаваршович Меликян – член-корреспондент Национальной академии наук РА, старший член IEEE,
зав. кафедрой, д.т.н., профессор
ORCID: 0000-0002-1667-6860; SPIN-код: не представлен
Элен Суреновна Казарян – аспирант, инженер-программист
SPIN-код: не представлен
Ашот Геворгич Арутюнян – д.т.н. профессор
SPIN-код: не представлен

Поступила в редакцию 21.04.2025
Одобрена после рецензирования 13.05.2025
Принята к публикации 26.05.2025