www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 18(1), 2027, 140-153. ISSN: 2007-1558.
https://doi.org/10.61467/2007.1558.2026.v17i6.1484

An Interactive Computational Tool Using the Gauss-Seidel Method for Power Flow Analysis in Electrical Engineering Education

Adael Amaury Ángeles López [1], Carlos Daniel Zamora Pérez [1], Julio Noel Pacheco Serrano [1], Gilberto Ortega Mendez [1], Javier Hernández Perez [2], Mario Oscar Ordaz Oliver [1].

[1] Departamento de Ingeniería Eléctrica y Electrónica, Tecnológico Nacional de México, Campus Pachuca, México.

[2] Departamento de Ingeniería Mecatrónica, Universidad Politécnica de Pachuca.

l22200580@pachuca.tecnm.mx, l22200629@pachuca.tecnm.mx, l22200615@pachuca.tecnm.mx, gilberto.om@pachuca.tecnm.mx, jahdez@upp.edu.mx, mario.oo@pachuca.tecnm.mx.

✉Corresponding author: l22200580@pachuca.tecnm.mx

Abstract.
The development and implementation of an interactive computational tool in MATLAB for power flow analysis is presented. The software addresses the challenges of modeling electric power systems under complex topologies and incomplete data. By applying the iterative Gauss-Seidel method, which optimizes computational load by avoiding explicit inversion of complex matrices and maintains robustness in small- and medium-scale networks, the algorithm automatically constructs the system's admittance matrix Ybus to obtain network parameters. Likewise, it processes the corresponding data for PQ load buses, PV controlled voltage buses, and Slack compensation buses, and integrates real-time visualization features that include single-line diagrams, convergence profiles, and error evolution metrics. Through validation by an n x n bus test system applied to 5 buses, the algorithm successfully converged in 43 iterations with a tolerance of $1x10^{-6}$, demonstrating stability in voltage magnitudes. Through the graphical interface that allowed direct parameter input, selection of per-unit or real values, and automatic topology rendering, operating as a digital twin for network evaluation. The results confirm that this tool not only improves the reliability of calculations but also provides a transparent, real-time view of the system state and its convergence behavior.
Keywords: Power flows, Ybus, Zbus, Electric Power Systems, analysis, iterations, software, interface.

Article Info
Received: Jun 16, 2026
Accepted: Aug 01, 2026

1 Introduction

AC power flow analysis is the cornerstone of power system steady-state studies, providing the foundation for planning, operation, optimization, and security assessment. While power flow analysis is a critical component in the future planning and structural expansion of Electric Power Systems, the mathematical models governing these networks form highly complex, non-linear systems. Consequently, obtaining a solution through traditional, closed-form analytical methods is highly impractical; such approaches suffer from slow computational speeds and a high risk of systemic collapse before reaching convergence. Furthermore, without an interactive computational topology to process complex data evaluating systemic equilibrium becomes inefficient. Therefore, the central problem addressed in this work is the need for a robust, user-interactive computational algorithm that utilizes iterative numerical methods, specifically Gauss-Seidel, to reliably calculate non-linear state variables and generate a visual digital twin of the network topology, preventing simulation failures and optimizing grid evaluation.

The Gauss–Seidel (GS) method has long been one of the foundational iterative techniques for solving the non-linear AC power flow problem. Its main advantages are simplicity of implementation, direct operation in the complex domain, and no requirement for Jacobian matrix inversion. A key modification to the GS method was introduced by Teng (Teng, 2002), who developed a modified Gauss–Seidel algorithm that separates a three-phase distribution network into three single-phase networks, solving them phase-by-phase. This is achieved by rearranging the admittance matrix according to phase relations and applying the implicit Z-bus superposition principle. The key innovation is that the LU factorization only needs to be performed once for three sub-matrices (YAA, YBB, YCC) rather than for the full Y-bus. It is important to implement

Gauss-Seidel because it has the advantage of significantly reducing computation time compared to manual or semi-automated methods. By taking advantage of MATLAB's ability to perform matrix operations and iterative calculations efficiently, it is possible to obtain results in seconds, even in systems with multiple nodes. For instance, it has been demonstrated that the Gauss-Seidel algorithm in MATLAB can compute more than a thousand iterations in less than one second (Hernández Jiménez et al., 2025). Likewise, this is verified when performing 'n' iterations until reaching a minimum error (Hernández Martínez et al., 2025). For a 270-bus system, this approach was six times faster than the implicit Z-bus method and 220 times faster than the conventional GS method, while maintaining virtually identical converged voltage solutions (Teng, 2002). Montoya and Gil-González (Montoya & Gil-González, 2020), proposed a successive approximation method for AC distribution networks that works directly in the complex plane, requires no derivatives, and only needs a single matrix inversion stored and reused across all iterations. For a 69-buses radial distribution system, this approach reached convergence in 5 iterations at 7.07 ms, while the classical GS method required 49,031 iterations. The method is proven to converge by the Banach fixed-point theorem under well-defined voltage conditions. Furthermore, it handles both radial and mesh topologies without any modifications to its mathematical structure, and supports multiple slack buses (Montoya & Gil-González, 2020). A fundamentally different line of modification replaces the classic GS update formula with an iteratively refined linear approximation based on Taylor series expansion. Bocanegra, Gil-González, & Montoya (2020), proposed a method that linearizes the hyperbolic voltage–power relation (1/V) using a first-order Taylor expansion around an operating point, then recursively updates that linearization point through successive iterations. This approach eliminated the estimation errors (up to 10%) present in earlier single-shot linear approximations, works with admittance matrices, making it independent of grid topology (radial or mesh), and was shown to have identical numerical behavior to Newton–Raphson. A related formulation was also extended to DC power grids by Montoya et al. (2019) where the successive approximation approach reduced processing time to just 0.46% of that required by the conventional GS method for a 21-buses test system, while achieving comparable accuracy (error below 1×10^{-8} %). Messalti et al. (2012) extended the GS modification to integrated AC–DC power systems containing HVDC links. The approach uses a sequential modified Gauss and Gauss–Seidel method where the DC system is treated via current injections at the converter buses, and the AC and DC equations are solved separately in each iteration, with boundary conditions matched iteratively. The key modification to the GS voltage update formula is the inclusion of DC current injection terms (Id) at converter buses. The method converged in 16 iterations for the modified IEEE 9-bus test system with an embedded HVDC link (compared to 21 iterations for the Gauss method). An earlier but important modification by Moorthy et al. (1995), introduced the reduction and restoration technique to the phase-coordinate GS algorithm. The method partitions buses into P-V (generator) and P-Q (load) types using matrix partitioning, then carries out the iteration cycle only on the reduced system (P-V buses). A restoration step updates the P-Q bus voltages after each iteration. For a balanced test system, this Improved Gauss–Seidel (I-G-S) method reduced iterations from 100 to 7, with CPU time decreasing from 2.08 s to 1.48 s (Moorthy et al., 1995). The work by Dilek et al. (Dilek et al., 2010), on sweep-based multiphase power flow introduced a load-stepping technique. When break points are introduced to radialize a meshed network, radial subsystems may become overloaded to the point where iterative solvers (including GS-like sweeps) diverge. The load-stepping approach starts with loads scaled to 0.1%, obtains an initial converged solution, then steps loads to 5% and 100%, using extrapolated current injections from previous steps as initial conditions. This technique made it possible to solve heavily loaded, heavily meshed distribution networks where traditional methods failed.

To address these computational limitations, it is hypothesized that integrating an iterative Gauss-Seidel mathematical engine within an interactive, SCADA-like digital twin will significantly improve the reliability and visualization of steady-state power flow analyses. By dynamically linking the mathematical resolution of non-linear state variables to a graphical topology, the risk of simulation failure due to poor initial estimations or data corruption can be mitigated, for this purpose, specific examples such as the use of SCADA in power systems show that it is designed so that this system can transfer all necessary data to the system operator, and provide specialists with the necessary information about the network and transformer substations in subregions, such as voltage, current, and productive capacity, in order to reduce network losses and help develop plans to manage energy distribution fairly (AbuMeteir, 2012). Therefore, it is necessary that the implementation of the solution also corrects to achieve what some authors mention: load flow studies were performed on calculating boards, which provided small-scale single-phase replicas of real systems by interconnecting circuit elements and voltage sources. Setting up the connections, making adjustments, and reading data was tedious and time-consuming. Digital computers now provide the solution to load flow studies in complex systems (Amlabu, 1999). And even though Gauss-Seidel processes more iterations, it carries out a process of selection and elimination while using the mathematical model in situations such as irregular and sparse matrices from electrical power system applications (Koester et al., 1994). In turn, some employ communication methods such as a parallel Gauss-Seidel solver implemented on the Thinking Machines CM-5 distributed memory multiprocessor (Koester et al., 1994), or those that implement combined AC-DC equations solved separately using sequential modified Gauss and Gauss-Seidel methods (Messalti et al., 2012). An important consideration for the selection of Gauss-Seidel is that it is more suitable for smaller and simpler power systems due to its computational efficiency, while the NR method provides superior accuracy and is more appropriate for large-scale systems (Samburi & Nor, 2025). Others add the implementation of SmartGrid, a method that proves to be effective for analyzing small power systems, providing accurate results with minimal computational complexity. The study demonstrates the importance of load flow analysis for understanding power system behavior and optimizing its operations (Hussain et al., 2024).

Therefore, this paper proposes the development of a custom, interactive computational tool implemented in MATLAB. The proposed software automatically constructs the system's admittance matrix and evaluates real and reactive power across load, controlled voltage, and compensation buses. To validate the robustness and operational equilibrium of the algorithm, the software is tested on a 5-bus electrical network, generating real-time single-line diagrams, convergence profiles, and error evolution metrics to provide a comprehensive evaluation of the power system's topology.

2 Methodology

When applying the equations, to solve the power flows, the use of admittances is considered whether self or mutual, that form the Ybus matrix. When solved, the Zbus matrix is obtained. To understand mathematical behavior, the numerical values are used for the series impedance Z and, for the load case, the line admittance Y.

$$Y_{ij} = |Y_{ij}|\angle\theta_{ij} \tag{1}$$

And this is represented as follows.

$$|Y_{ij}|\text{cos}\theta_{ij} + j|Y_{ij}|\text{sen}\theta_{ij} \tag{2}$$

For the case where machines such as transformers are placed, the nominal values, impedances, and taps are added, since in power flow cases these data influence. Also, the bus voltage must be given, and this leads to a bus with a considerable voltage, represented in polar coordinates as:

$$G_{ij} + jB_{ij} \tag{3}$$

Given the existence of voltage, it is represented mathematically as:

$$V_i = |V_i|\angle\delta_i \tag{4}$$

Solved in the following way:

$$V_i = |V_i|\angle\delta_i\ |V_i| = (\cos\delta_i + j\sin\delta_i) \tag{5}$$

For the behavior of the current that is inside the network and related to the Ybus matrix, it is:

$$I_i = Y_{i1}V_1 + Y_{i2}V_2 + \cdots + Y_{in}V_n \tag{6}$$

For the effect represented in real power and reactive power that are being added to the network, their complex conjugate is:

$$P_i - jQ_i = V_i^* \sum_{n=1}^{n} Y_{in}\, V_n \tag{7}$$

And it is by applying magnitudes and angles:

$$P_i - jQ_i = \sum_{n=1}^{n} |\, Y_{in}V_iV_n|\angle(\theta_{in} + \delta_n - \delta_i) \tag{8}$$

Demonstrating that the real and reactive parts result in:

$$Q_i = -\sum_{n=1}^{n} |Y_{in}V_iV_n| \sin(\theta_{in} + \delta_n - \delta_i) \quad \textbf{(9)}$$

$$P_i = \sum_{n=1}^{n} |Y_{in}V_iV_n| \cos(\theta_{in} + \delta_n - \delta_i) \quad \textbf{(10)}$$

For practical purposes, it is considered that the previous equations (9) and (10) form the polar coordinates part within the power flows, since they inject into the network at a certain bus a value which, considering that power as Pgen,i is generating at a selected bus and the case of Pdem,i is considered as power that is understood to be part of the load. From this, the scheduled injected power is obtained into the network at the considered bus, and in practical methods it is called:

$$P_{i,\text{prog}} = P_{\text{gen},i} - P_{\text{dem},i} \quad \textbf{(11)}$$

And with equation (11), the power added to the software system is presented to the software system, and it is indicated as total power at the selected bus, allowing us to know the power behavior through Figure 1.

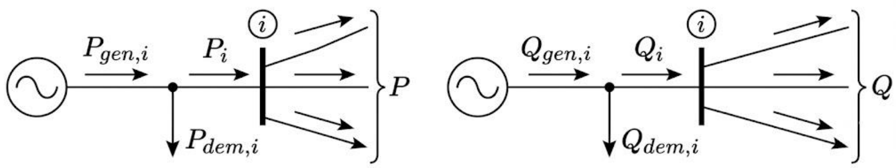


Fig. 1. Real and reactive power at the selected bus.

Within the study to find the value of Pi and understand it as Pcalc, i the error is obtained as the relationship of Pprog, i less the calculation of Pcalc, i. Where the real power results from the subtraction of the scheduled power, which includes the scheduled power generated and the demand power at the bus.

$$\Delta P_i = P_{\text{prog},i} - P_{\text{calc},i} = \left(P_{\text{gen},i} - P_{\text{dem},i}\right) - P_{\text{calc},i} \quad \textbf{(12)}$$

And the reactive power results from the subtraction of the scheduled power, which includes the scheduled generated power and the demand power at the bus.

$$\Delta Q_i = Q_{\text{prog},i} - Q_{\text{calc},i} = \left(Q_{\text{gen},i} - Q_{\text{dem},i}\right) - Q_{\text{calc},i} \quad \textbf{(13)}$$

To prevent the power flow calculation program from being invalidated by error, it is considered that both equations (12) and (13) are equalized in their calculated part; it is considered that the error reaches zero within the bus and they are balanced by means of:

$$g_{i'} = P_i - P_{\text{prog},i} = P_i - \left(P_{\text{gen},i} - P_{\text{dem},i}\right) = 0 \quad \textbf{(14)}$$

$$g_{i''} = Q_i - Q_{\text{prog},i} = Q_i - \left(Q_{\text{gen},i} - Q_{\text{dem},i}\right) = 0 \quad \textbf{(15)}$$

Therefore, it can be considered that if, once the bus information is loaded into the software, a situation occurs where one or several values have not been added, they are simplified and the solution system nullifies them as a method to prevent corrupting the practical application at startup. This allows the rest of the test to be solved without risking the data and considerations for the selected power flow problem. It is understood that for a bus there are three types of data loading to choose its behavior in the system, and for the case where there is the possibility of adding 4 data but only two are placed, so that the following are calculated by means of the rest of the system.

$$\delta_i \quad \textbf{(16)}$$

$$|V_i| \quad \textbf{(17)}$$

$$P_i \quad \textbf{(18)}$$

$$Q_i \quad \textbf{(19)}$$

Which are identified as equation (16) for phase angle, equation (17) for phase magnitude, equation (18) for real power, and equation (19) for reactive power.

Applying the 3 methods resulting:

Load Bus:

We consider these points where real power and reactive power are loaded and programmed since they are part of the consumption area and are known data. The total consumption at the moment is known, and within this data load, both phase magnitude and angle remain pending with respect to the programming, and the system process solves all the remaining data.

Controlled voltage bus:

These are analyzed as the buses that control or are controlled by a generator in a stable manner. Here, within the programming, information on the generated real power is available and the phase magnitude of the voltage injection. Here, voltage turns out not to be an unknown; however, information on the system's reactive power is not available, which indicates that the generator is forced to maintain its voltage to avoid failures in the power system. Within the software, it initially obtains the phase angle to obtain the required reactive power.

Compensation Bus:

Also known as Slack or the reference bus, and it is practically assigned a stable voltage and a reference angle, which is usually (0°) in most cases. This works as a way to justify the results, since in a real system there are failures, losses, and this bus is used to maintain the system, being able to add or subtract energy that may be lost. Within programming, real power and reactive power data are not loaded, and within the software, when it reads that this bus is a reference, it does not apply its solution; only once everything has been obtained does it proceed to the solution.

Applying the Law of Conservation of Energy, which indicates that the total energy injected by the generators at the buses must be compensated by the considered loads, adding losses due to temperature issues or effects occurring at the buses, transmission lines, and physical equipment in the substation. Therefore, for Pi and Q at any bus of the system, it relates to equation (14), since i can be interpreted by a considered quantity n, and by adding the n number of programmed system equations, it is obtained.

$$\begin{aligned} P_L &= \text{real power loss} \\ P_L &= \sum_{i=1}^{n} P_{\text{gen},\,i} - \sum_{i=1}^{n} P_{\text{dem},\,i} \end{aligned} \quad \textbf{(20)}$$

Since Pi becomes the losses in the entire system (transmission lines and transformers), within the software system the solution to obtain the individual currents becomes a final process because it is necessary to know the magnitude and angle of the voltage at each bus of the electrical power system.

There is a situation within the power flow solution that occurs when taking a bus such as the compensation bus where the Pgen programming has not yet been done. There is a compensation difference due to the total P of the system when it is connected to the system from all buses and the output of P, and by adding the losses, the assignment to the compensation bus occurs once the total result has been printed. Within the software, a generator bus with slack is internally selected because the total megavars supplied by the generators and the megavars received at the loads are related.

$$\sum_{i=1}^{n} Q_i = \sum_{i=1}^{n} Q_{\text{gen},\,i} - \sum_{i=1}^{n} Q_{\text{dem},\,i} \tag{21}$$

Losses are given by two possible forms: losses in power from equation (22) and losses in transmission lines given by equation (23).

$$I^2R \tag{22}$$

$$I^2X \tag{23}$$

To understand how the methods are applied in the system, it is vital to know that both the voltage magnitudes and angles that have not been programmed serve as state or dependent variables, since their values represent the system and its state, and they depend on the quantities entered at the buses. Through power flows, state values are determined to solve the implemented equations. Therefore, the relationship between Pgen (excluding the slack) within a system of n buses will have the same number of equations to solve as state variables to find and apply, because once found, the complete state of the electrical power system is known. Hence, the quantities from equations (18) and (19) within the slack, or equation (19) in controlled voltages, and the case of losses through equation (18) are independent functions. Since equations (18) and (19) are nonlinear functions of the state variables from equations (16) and (17), iterations are applied using the Gauss-Seidel procedure.

Applying the Gauss-Seidel method, it must be understood that the solution of an electrical system by itself and obtaining a result through power flows is complex because there are differences between the data and considerations such as bus types or what data is available. Obtaining the solution through a closed system is not practical because it is not fast and could cause the system to collapse before reaching an answer. Therefore, Gauss-Seidel applies an iterative process using estimated values for the unknown bus voltages and, through the process, calculates bus by bus, using estimates from other buses and data such as real power and reactive power.

Thus, by carrying out this process, values for each bus are obtained, which are then used to obtain another bus as a solution order. Each completed process is called an iteration, and it stops when the changes in the buses have been reduced to a minimum, as explained below.

The slack bus is designated as bus 1, and the system continues with buses 2, 3, and up to n buses. If P and Q are already programmed in the software, they are obtained.

$$\frac{P_{\text{prog},\,2} - jQ_{\text{prog},\,2}}{V_2^*} = Y_{21}V_1 + Y_{22}V_2 + Y_{23}V_3 + Y_{2n}V_n \tag{24}$$

Consequently, it is concluded that

$$V_2 = \frac{1}{Y_{22}}\left[\frac{P_{\text{prog},\,2} - jQ_{\text{prog},\,2}}{V_2^*} - (Y_{21}V_1 + Y_{23}V_3 + Y_{2n}V_n)\right] \tag{25}$$

Therefore, when considering buses 3, 4, or n buses with specified real and reactive power loads, they are applied as follows.

$$V_3 = \frac{1}{Y_{33}}\left[\frac{P_{\mathrm{prog},3} - jQ_{\mathrm{prog},3}}{V_3^*} - (Y_{31}V_1 + Y_{32}V_2 + Y_{3n}V_n)\right] \quad \textbf{(26)}$$

And since the methods are similar, and in turn the real and imaginary part can lead to more equations because it considers the application of iterations to find the complex voltages of the equations, it works through the mechanism of powers, deriving this process.

$$V_1 = |V_1|\angle\delta_1, \qquad V_2^{(0)}, \quad V_3^{(0)}, \quad V_4^{(0)} \quad \textbf{(27)}$$

Given that V2 through Vn are the assigned buses. And applying to correct the voltage, it is represented as.

$$V_2^{(1)} = \frac{1}{Y_{22}}\left[\frac{P_{\mathrm{prog},2} - jQ_{\mathrm{prog},2}}{V_2^{(0)*}} - \left(Y_{21}V_1 + Y_{23}V_3^{(0)} + Y_{2n}V_n^{(0)}\right)\right] \quad \textbf{(28)}$$

Therefore, each time a correct voltage is found at the assigned bus, that value is used to calculate the next one in order and substitutes into the equation, occurring as follows.

$$V_3^{(1)} = \frac{1}{Y_{33}}\left[\frac{P_{\mathrm{prog},3} - jQ_{\mathrm{prog},3}}{V_3^{(0)*}} - \left(Y_{31}V_1 + Y_{32}V_2^{(1)} + Y_{3n}V_n^{(0)}\right)\right] \quad \textbf{(29)}$$

If in the case it is repeated again according to the n buses, except for the slack bus, until reaching the iteration where it is reduced until obtaining a precise value, this avoids convergence above an error of values that act as phase magnitude which does not change considerably, and therefore the initial estimates of the voltages are equally applied to the loads as follows.

$$1.0\angle 0^\circ \quad \textbf{(30)}$$

This is in per unit, and the applicable equation for any n bus of the system by quantity, where both real and reactive power are programmed, is represented as.

$$V_i^{(k)} = \frac{1}{Y_{ii}}\left[\frac{P_{\mathrm{prog},i} - jQ_{\mathrm{prog},i}}{V_i^{(k-1)*}} - \sum_{j=1}^{i-1} Y_{ij}\,Y_j^{(k)} - \sum_{j=i+1}^{n} Y_{ij}\,Y_j^{(k-1)}\right] \quad \textbf{(31)}$$

The superscript (k) indicates the iteration number at which the voltage has been calculated, and (k – 1) indicates the one preceding it in the iteration.

3 Experimental procedures

3.1 Software design

Since the modeling of the electrical power system requires the application of software for its solution, MATLAB is used as the system because its configuration allows the user to impart and implement new information, as well as software updates based on regulatory mentions or the needs requested by clients or designers. Therefore, within the architecture it is considered for event simulation since it is interactive with the user and suggests activities through the implementation of windows for selecting and operating multiple commands.

3.1.1 Configuration of use

Once the system has started, one must understand the loading medium for the data, the application of methods in an academic or real-world context with data such as per-unit or real unit formats, and also the system configuration must determine its convergence tolerance index and acceleration, and other cases for the amount of iteration time by means of mathematics.

3.1.2 Data reception

By feeding information into the system, adding matrices, as well as generation and load data in the system such as real power and reactive power, phase angles and magnitudes, operational limitations, the system is allowed to understand the behavior and thus apply the mathematical method to obtain results quickly with respect to the applied configuration.

3.1.3 Visualization

Once the system has been selected to apply the solution, mathematics begins to solve it. Its response, having an interactive window, allows the user to host various viewing methods, from the presentation of result tables, the solution to pre-loaded matrices, and the design of the topology, all related to the convergence of the error with respect to the applied method.

3.2 Application of Matrix Topology

3.2.1 Ybus and Zbus

Since power flows are applied within the implemented software, the addition of matrices is required, and this determines that within the Ybus matrix, complex data is directly loaded, and in turn, by applying the inverse form in the solution, the Zbus is obtained. An internal mathematical application is the use of Kirchhoff's Laws for the construction, in other cases, of the Ybus, because if some values are found to be negative or different in the symmetry of the matrix, the solution applies to put the data in order, such as placing negative values off the diagonal.

3.3 Algorithm and Mathematical Engine

Within the system, a mathematical approach has been considered for solving power flows, so this medium applies the methods for solution using Gauss-Seidel. Therefore, within the simulation software, a solution is applied using nonlinear algebraic equations, and through Gauss-Seidel, it seeks an operating equilibrium by iteration, as indicated by applying equation (1). When applying the method, it is considered to be a closed loop, since this allows it to generate the model of the real power and reactive power buses by applying equations (9) and (10).

3.4 Rendering

A system is applied within the software so that when the error appears and crosses the tolerance and leads the system to converge, it applies a visualization of the reached point and displays it, applying graph theory, constructing through the data. Likewise, for the Ybus matrix and its solution, it draws digital connections within a window. The bus layout is reflected avoiding simulation software failures, and it is also a digital twin of systems such as SCADA, since it allows data evaluation, presents a single-line diagram with respect to voltage magnitude, phase angle, and the bus classifier that is influenced by being a bus corresponding to the reference, a generation bus, or a load bus.

4 Results and discussion

In this section, the results obtained through the use of the software are presented made in MATLAB. The development implements the creation of the model and topology of the system. The set of information applied in the set of information applied in each section by means of tables, presenting the evaluation.

4.1 Interface

To solve the problem, the software implemented using MATLAB must be evaluated This presents the windows to work with. Within the main window, the area containing sections is shown: the first to indicate the values, either as shown in per unit or through real values; the second space is to indicate the inputs, which allows, through two sub-windows, the selection of information to add, one for matrices in the case of Ybus, Zbus or line parameters inserted individually, and for Bus Data the values are placed in per unit and this is related to the data of equations (16-19); and the third section is System Topology and Results, which integrates the printing of the results once they have been loaded and the program has been run. It provides us with a graphical and user-friendly way, through the Final Buses Report, adding the case of the line as a Single-Line Diagram, the Convergence Profile and Error Evolution, which are related to the Gauss-Seidel mathematical method.

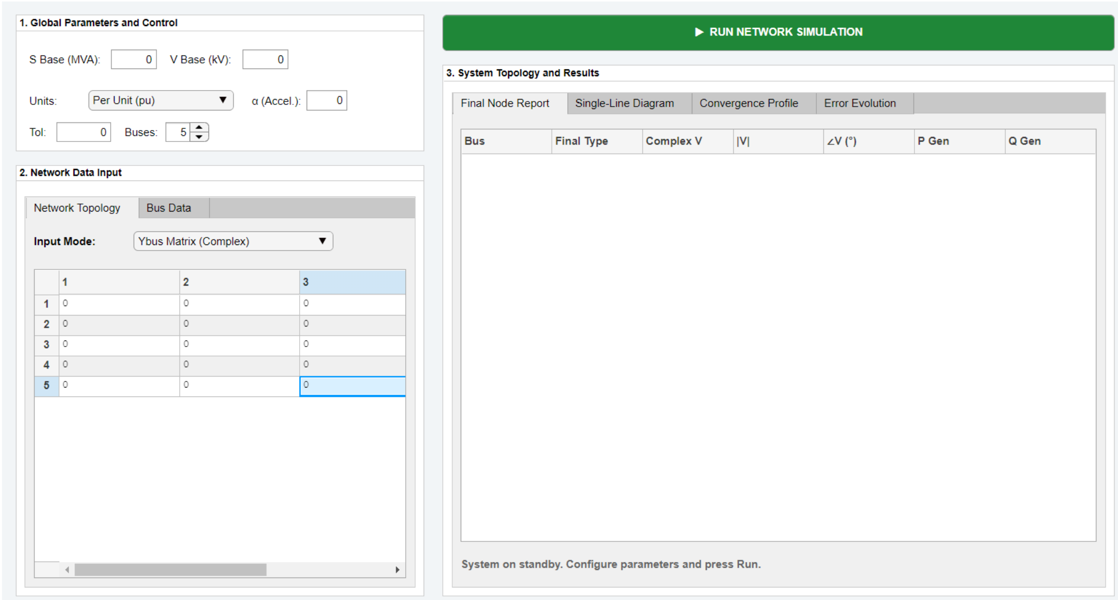


Fig. 2 Work Interface.

The data presented for the solution through the interface are the following:

4.2 Description

Taking as a base model, as well as the applicable theory in the interface obtained from (Glover et al., 2011). In this section, the proposed exercise is explained, as well as the process using the interface. First-hand, the data is included along with which section it corresponds to, and the single-line topology of the system corresponds to the following figure:

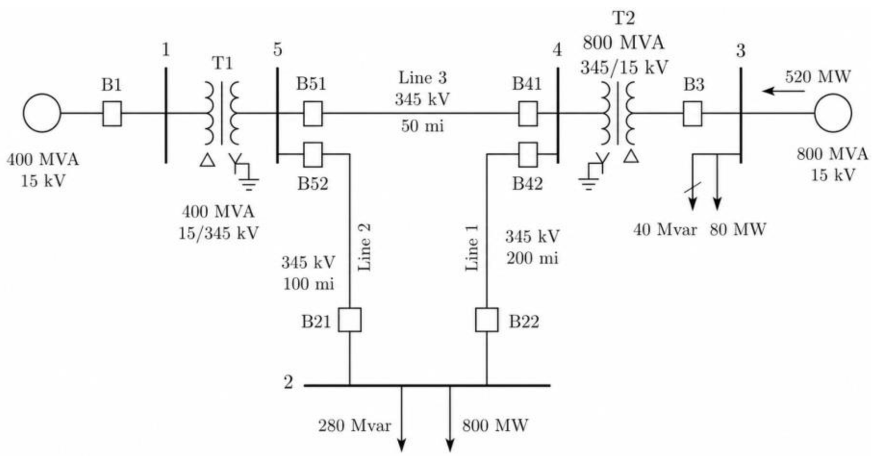


Fig. 3 Single-line system topology with values.

Related information to add to the interface about the buses.

Table 1. Input to the system buses.

Bus	Type	V_{pu}	δ	$P_{G\ pu}$	$Q_{G\ pu}$	$P_{L\ pu}$	$Q_{L\ pu}$	$Q_{Gmin\ /\ Gmax\ pu}$
1	Slack	1.0	0	-	-	0	0	-
2	PQ	-	-	0	0	8.0	2.8	-
3	PV	1.0	-	5.2	-	0.8	0.4	-2.8 and 4.0
4	PQ	-	-	0	0	0	0	-
5	PQ	-	-	0	0	0	0	-
S_{base}	100 MVA							
V_{base}	15 kV at 1 and 3							
V_{base}	345 kV at 2, 4 and 5							

Table 2. Inputs to transmission lines.

Bus to Bus	R'_{pu}	X'_{pu}	G'_{pu}	B'_{pu}	$MVA_{max\ pu}$
2 – 5	0.0090	0.100	0	1.72	12.0
2 – 4	0.0045	0.050	0	0.88	12.0
4 – 5	0.00225	0.025	0	0.44	12.0

Table 3. Input information for the transformer.

Bus to Bus	R_{pu}	X_{pu}	$G_{c\ pu}$	$B_{m\ pu}$	$MVA_{max\ pu}$	TAP_{pu}
1 – 5	0.00150	0.02	0	0	6.0	-
3 – 4	0.00075	0.01	0	0	10.0	-

Once the data from the 3 tables is contained, it is entered into the program where the admittance matrix corresponding to the buses and lines that connect them is loaded:

$$\begin{bmatrix} 3.73-j49.72 & 0 & 0 & 0 & -3.73+j49.72 \\ 0 & 2.68-j28.46 & 0 & -0.89+j9.92 & -1.79+j19.84 \\ 0 & 0 & 7.46-j99.44 & -7.46+j99.44 & 0 \\ 0 & -0.89+j9.92 & -7.46+j99.44 & 11.92-j147.96 & -3.57+j39.68 \\ -3.73+j49.72 & -1.79+j19.84 & 0 & -3.57+j39.68 & 9.09-j108.58 \end{bmatrix} \quad \textbf{(32)}$$

Within the program, it has been selected that the Vbase used is 345 kV since the corresponding Vbase is 15 kV and it is only between the generator and transformer. Therefore, when evaluating the program we use the highest value because we are studying the behavior within the lines for the power flow.

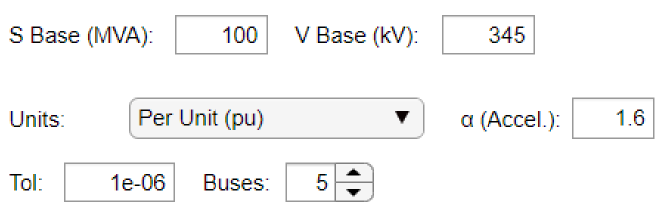
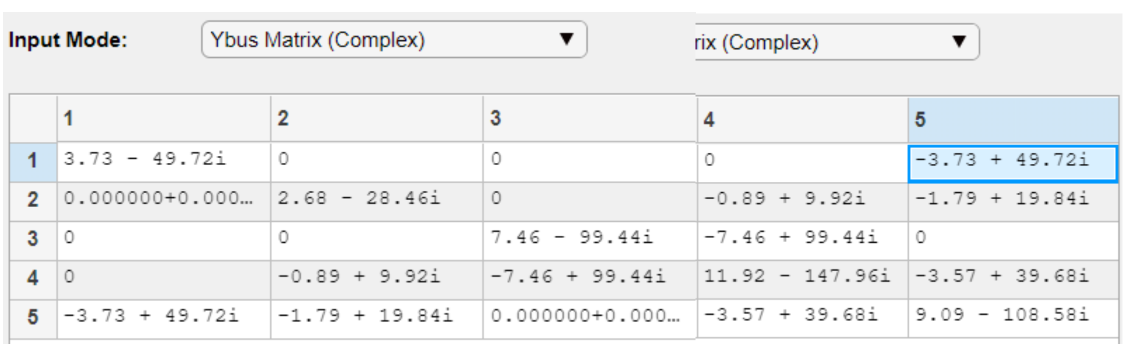


Fig. 4 Global Parameters and Control

In Fig. 4, the global and control parameters for the simulation of the 5-bus power system are shown, establishing the power in relation to 100 MVA, the main voltage of 345 kV, operating in per unit, which allows simplifying the model. For the solution, the algorithm was configured with a convergence tolerance of 1x10-6 and an acceleration factor corresponding to 1.6.



	1	2	3	4	5
1	3.73 - 49.72i	0	0	0	-3.73 + 49.72i
2	0.000000+0.000…	2.68 - 28.46i	0	-0.89 + 9.92i	-1.79 + 19.84i
3	0	0	7.46 - 99.44i	-7.46 + 99.44i	0
4	0	-0.89 + 9.92i	-7.46 + 99.44i	11.92 - 147.96i	-3.57 + 39.68i
5	-3.73 + 49.72i	-1.79 + 19.84i	0.000000+0.000…	-3.57 + 39.68i	9.09 - 108.58i

Fig. 5. Ybus matrix construction

Within the construction of the matrix, the design is that, having a customized menu, a bar is used to navigate through the system and view all the information. Also, being an easy-access model, it does not require the information to be complex to load.

This matrix is obtained from the admittances by obtaining them from the bus and line data with the aforementioned information, also in relation to Fig. 3.

Now, the Bus Data section contains the information related to the buses and is presented below in the figure.

Network Topology | Bus Data

		Type	P Gen (pu)	Q Gen (pu)	Q Min (pu)	Q Max (pu)	P Load (pu)	Q Load (pu)	|V| (pu)	Initial ∠
1	1	Slack	0	0	0	0	0	0	1	0
2	2	PQ	0	0	0	0	8	2.8000	1	0
3	3	PV	5.2000	0	-2.8000	4	0.8000	0.4000	1	0
4	4	PQ	0	0	0	0	0	0	1	0
5	5	PQ	0	0	0	0	0	0	1	0

Fig. 6. Bus Data integrating information.

Once the system has been started to find the answer, it shows us this:

✓ Simulation successfully completed in 43 iterations. Format: Per Unit (pu)

Fig. 7. "Problem Resolved" indicator

It indicates that the Gauss-Seidel method has completed the exercise, applying up to 43 iterations to reach the target point.

Now, once that is mentioned, a table called Final Buses Report is printed, which integrates the solution information for the system for each bus in order from 1 to 5, including the bus type that mentions its dynamics in the mathematical model, voltage indicators for magnitudes, and includes the real output power and reactive output power.

Final Node Report | Single-Line Diagram | Convergence Profile | Error Evolution

	bus	dynamic_types	V_complex_str	V_mag_out	V_ang_final_deg	PG_out	QG_out
1	1.0000	Slack	1.000000+0.000000i	1.0000	0	3.9988	2.5535
2	2.0000	PQ	0.706895-0.322208i	0.7769	-24.5039	0	0
3	3.0000	PV	0.999997-0.002514i	1.0000	-0.1440	5.2000	2.7944
4	4.0000	PQ	0.972646-0.044650i	0.9737	-2.6283	0	0
5	5.0000	PQ	0.942930-0.076145i	0.9460	-4.6168	0	0

Fig. 8. Final Report.

Furthermore, when solving it, more windows with important results are included, which will be presented in order by their location within the System Topology and Results section.

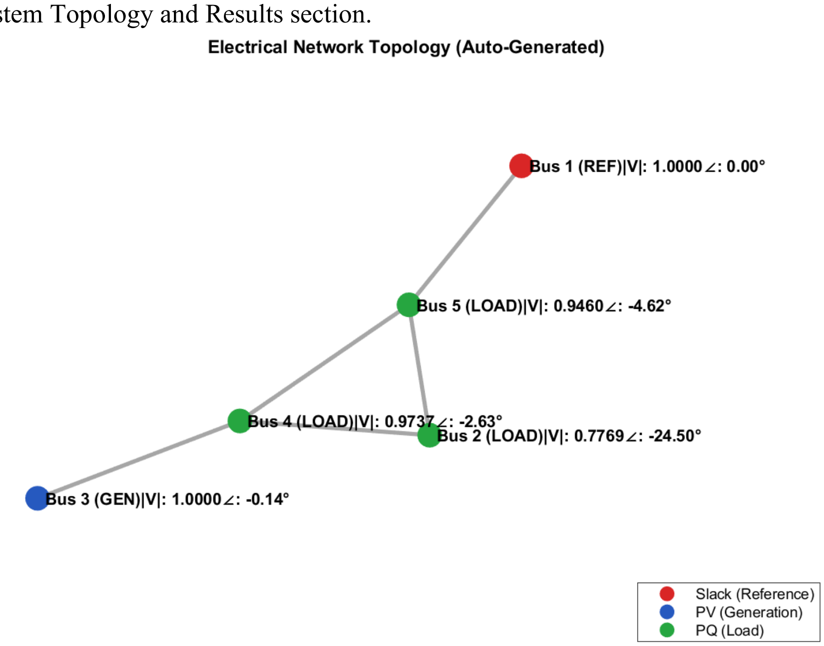


Fig. 9. Single-Line Diagram.

Using the data loaded into the program, it builds the single-line diagram as well as the type of bus corresponding to the information, their values in per unit, and emphasizing the inclusion of the angles. This program has the capability to automatically generate said graph.

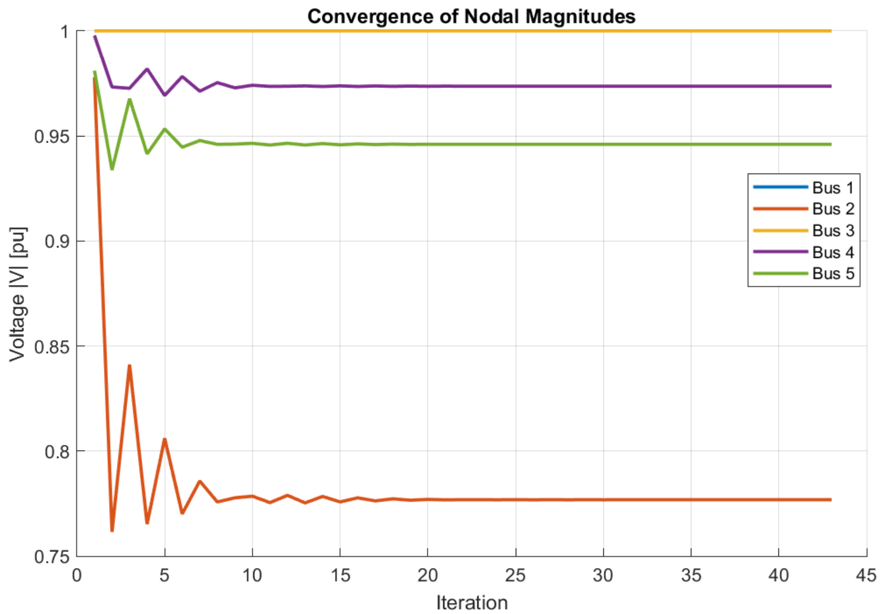


Fig. 10. Convergence Profile.

In figure 10, the evolution of the error and its relationship to the convergence profile for the 5-bus system is presented, adding that it is shown with voltages in per unit. The effectiveness of the Gauss-Seidel method is demonstrated by applying the mathematical model, starting from the initial conditions where its behavior is stable. The algorithm begins by adjusting the magnitudes of the buses that are considered for load reactive power. For the fluctuations within the first iterations, this shows the corrections against the error in the calculation of the injected powers. Once it approaches 25 iterations, the convergence profile has been successfully completed, and once the program stops at 43 iterations, this is due to the time applied for the solution.

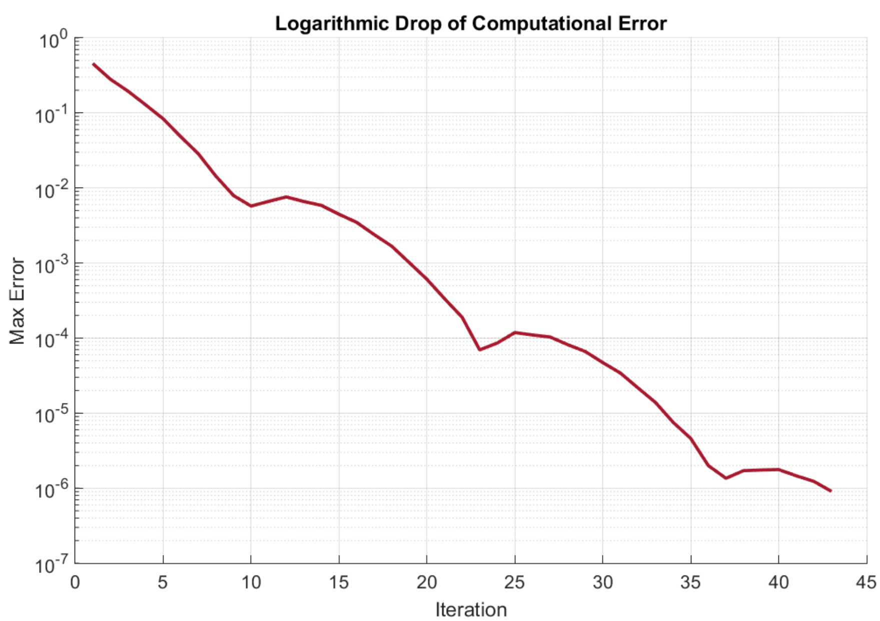


Fig. 11. Error Evolution.

The analysis of the 5-bus system applying the Gauss-Seidel method starts from flat start initial conditions. The load buses, known as load reactive power, experience behaviors and are corrected once the program begins its operation. It is observed that the +problem stabilizes once 25 iterations are reached. In turn, the voltage buses maintain their magnitude in per unit at

1.0. Being a system solved by an algorithm, it shows that the error begins to drop and continues correcting details until 43 iterations, at which point the maximum error with respect to the predefined tolerance crosses the threshold. Thus, the solution concludes and the program's response is specified.

5 Conclusions

The implementation of the software developed in the MATLAB environment represents a significant improvement in power flow analysis compared to conventional computational methods. Through its graphical and interactive interface, the program optimizes parameter input, allowing direct loading of complex and real data for the construction of the Ybus matrix. Furthermore, it overcomes the visualization limitations of other tools by automatically generating the single-line topology of the system, acting as a digital twin that facilitates network evaluation.

A fundamental analytical advantage of this proposal is the ability it gives the user to dynamically visualize the mathematical behavior of the model in relation to iterations and the error margin. By applying the Gauss-Seidel method to solve the nonlinear algebraic equations, the system is not limited to delivering a closed result; on the contrary, it allows exhaustive monitoring of the convergence profile and error evolution.

This behavior is evidenced in the system simulation, where it is graphically observed how the algorithm corrects fluctuations in the calculation of injected powers during the first iterations. Thanks to these integrated graphs, it is possible to verify that the system reaches stability around iteration 25, and the analytical process with high precision at iteration 43 when it crosses the predefined tolerance threshold. In short, the developed tool not only accurately calculates magnitudes, angles, and powers, but also provides the operator with a robust and transparent environment for visual evaluation of the state and stability of the electrical system.

References

AbuMeteir, H. A. M. (2012). *A proposed SCADA system to improve the conditions of the electricity sector in Gaza Strip* [Master’s thesis, Islamic University of Gaza].

Amlabu, C. A. (1999). Application of computer for solutions of electrical load-flow studies in electrical power system [Project report]. Federal University of Technology Minna.

Bocanegra, S. Y., Gil-González, W., & Montoya, O. D. (2020). A new iterative power flow method for AC distribution grids with radial and mesh topologies. In *2020 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)* (pp. 1–5). IEEE. https://doi.org/10.1109/ROPEC50909.2020.9258750

Dilek, M., de León, F., Broadwater, R., & Lee, S. (2010). A robust multiphase power flow for general distribution networks. *IEEE Transactions on Power Systems, 25*(2), 760–768. https://doi.org/10.1109/TPWRS.2009.2036791

Glover, J. D., Sarma, M. S., & Overbye, T. J. (2011). *Power system analysis and design* (5th ed.). Cengage Learning.

Hernández Jiménez, R., López Portillo Zuarth, D., Díaz Ledezma, M. A., Ordaz Oliver, M. O., & Hernández Martin Yael, L. (2025, July). *Power flow analysis in MATLAB using the Gauss-Seidel algorithm* [Conference paper]. Reunión Internacional de Verano de Potencia y Aplicaciones Industriales (RVP-AI) y Reunión Internacional de Otoño de Comunicaciones, Computación, Electrónica, Automatización y Robótica (ROC&C), Acapulco, Mexico.

Hernández Martínez, A. M., Díaz Ledezma, M. A., López Portillo Zuarth, D., Melgarejo Tolentino, Y., & Ordaz Oliver, M. O. (2025). Análisis del rendimiento del algoritmo de Gauss Seidel para estudio de flujos de potencia mediante MATLAB. In *Memorias del Congreso Internacional de Mecatrónica, Control e Inteligencia Artificial (CIMCIA), Año 4, Tomo 4* (pp. 290–295). Universidad Nacional Autónoma de México, Facultad de Estudios Superiores Cuautitlán.

Hussain, M. D., Rahman, M. H., & Ali, N. M. (2024). Investigation of Gauss-Seidel method for load flow analysis in smart grids. *Scholars Journal of Engineering and Technology, 12*(5), 169–178. https://doi.org/10.36347/sjet.2024.v12i05.004

Koester, D. P., Ranka, S., & Fox, G. C. (1994). A parallel Gauss-Seidel algorithm for sparse power system matrices. In *Supercomputing ’94: Proceedings of the 1994 ACM/IEEE Conference on Supercomputing* (pp. 184–193). IEEE Computer Society Press. https://doi.org/10.1145/602770.602806

Messalti, S., Belkhiat, S., Saadate, S., & Flieller, D. (2012). A new approach for load flow analysis of integrated AC–DC power systems using sequential modified Gauss–Seidel methods. *European Transactions on Electrical Power, 22*(4), 421–432. https://doi.org/10.1002/etep.570

Montoya, O. D., & Gil-González, W. (2020). On the numerical analysis based on successive approximations for power flow problems in AC distribution systems. *Electric Power Systems Research, 187*, 106454. https://doi.org/10.1016/j.epsr.2020.106454

Montoya, O. D., Garrido, V. M., Gil-González, W., & Grisales-Noreña, L. F. (2019). Power flow analysis in DC grids: Two alternative numerical methods. IEEE Transactions on Circuits and Systems II: Express Briefs, 66(11), 1865–1869. https://doi.org/10.1109/TCSII.2019.2891640

Moorthy, S., Al-Dabbagh, M., & Vawser, M. (1995). Improved phase-coordinate Gauss-Seidel load flow algorithm. *Electric Power Systems Research, 34*(2), 91–95. https://doi.org/10.1016/0378-7796(95)00960-6

Samburi, N., & Mohamad Nor, A. F. (2025). A comparative study of Gauss-Seidel and Newton-Raphson methods in power flow analysis. *Multidisciplinary Applied Research and Innovation, 6*(2), 89–95. https://doi.org/10.30880/mari.2025.06.02.010

Teng, J.-H. (2002). A modified Gauss–Seidel algorithm of three-phase power flow analysis in distribution networks. *International Journal of Electrical Power & Energy Systems, 24*(2), 97–102. https://doi.org/10.1016/S0142-0615(01)00022-9