LOAD FLOW AND SHORT-CIRCUIT STUDY REPORT

A COMPARATIVE ANALYSIS USING MATLAB AND DIGSILENT POWERFACTORY

POWER SYSTEMS ANALYSIS

Prepared by:

Gift Rethabile Mthombeni

(BEng Tech Electrical Engineering)

AUGUST 2025

Table of Contents

1. Introduction

Load flow is an important tool used by power engineers for planning, to determine the best operation for a power system and exchange of power between utility companies. In order to have an efficient operating power system, it is necessary to determine which method is suitable and efficient for the system's load flow analysis

This report presents a comprehensive analysis of the IEEE 4-bus power system, focusing on load flow and short-circuit studies conducted using MATLAB (both Simulink and M-file scripting) and DIgSILENT PowerFactory. This study aims to assess system performance under steady-state and fault conditions, verify simulation results through manual calculations, and deepen understanding of power system fundamentals.

The Newton Raphson method is used to solve load flow analysis problems. The Newton-Raphson method is a more efficient iterative approach and converges faster. It solves the load flow equations simultaneously by linearizing the system equations around a starting assumption and then iteratively updating the bus voltages until convergence is achieved.

A three-phase symmetrical short-circuit study is conducted using the IEC 60909 standard, which provides a systematic approach to calculating short-circuit currents for equipment rating and protection coordination (CIGRE, 2014). The chosen base power of 150 MVA and base voltage of 132 kV are applied consistently to normalize system quantities.

A key outcome of this project is the validation of simulation results by comparing them with manual analytical solutions, thereby enhancing both theoretical knowledge and practical competence. This aligns with Graduate Attribute 4 (GA4), which emphasizes the ability to investigate and solve broadly defined engineering problems through the integration of theory, simulation, and computational tools (ECSA, 2019).

2. Identifying Broadly Defined Problems for Investigation

Identify the broadly defined problems related to power system analysis, focusing on the load flow problem using the Newton-Raphson method for the IEEE 4-bus system.

Problem Statement: Clearly articulate the challenges in solving load flow for the 4-bus system, such as ensuring convergence, handling nonlinear equations, or managing voltage stability. Note that the equations you include here are somehow computed in your M-File.

In power system analysis, the load flow problem constitutes a fundamental challenge, particularly in multi-bus systems such as the IEEE 4-bus network. The problem statement centres on solving nonlinear algebraic equations that describe the balance of real and reactive power at each bus. For the 4-bus system, this involves determining unknown bus voltages and phase angles while ensuring that specified power injections and withdrawals are satisfied.

A central challenge is the nonlinearity of the power flow equations:

$$P_i = \sum_{j=1}^{n} |V_i|\left|V_j\right| \left(G_{ij}\cos\theta_{ij} + G_{ij}\sin\theta_{ij}\right)$$

$$Q_i = \sum_{j=1}^{n} |V_i||V_j| \left(G_{ij} \sin\theta_{ij} + G_{ij} \cos\theta_{ij}\right)$$

Where Pi and Qi are the real and reactive power at bus i, Vi is the bus voltage, and, Gij, Bij are elements of the bus admittance matrix. These equations are typically solved iteratively using the Newton-Raphson method, which applies Jacobian-based linearization.

The broadly defined problems associated with solving this include:

- Convergence reliability: NR generally converges quickly, but may diverge or oscillate if the initial guess is poorly chosen (Glover et al., 2015).
- Computational complexity: While efficient for small systems, the method requires significant computational resources for very large-scale networks due to repeated Jacobian updates (Kundur, 1994).
- Voltage stability concerns: Near-critical loading conditions can cause ill conditioning of the Jacobian, complicating convergence.
- Integration into software tools: Translating these equations into MATLAB M-file coding requires accurate implementation of Y-bus construction and mismatch functions, while DIgSILENT automates this but still requires careful parameterisation.

The following two issues were of particular concern, that is, the ***reactive power limit*** and ***convergence*** test:

Given that this is a broadly defined problem, the reactive power limits for the generator bus 2 were not explicitly stated. Knowing the limits is crucial in conducting the *reactive power limit test* for a PV bus, to determine if it satisfies the criteria as outlined in the test or not. This test allows the confirmation of the PV bus as a PV bus if the criteria is satisfied or its classification as a PQ bus if the criteria is not satisfied, and this has overall implications on the number of power mismatches the system has, the size of the Jacobian matrix and subsequently the solutions.

Similarly, in the problem, the convergence tolerance level is not specified. The tolerance level is used to conduct the *convergence test* which gives a threshold with respect to the acceptable minimum allowable error for the converging solution.

Thus, the investigation addresses these challenges by comparing manual Newton Raphson solutions with MATLAB coding results and DIgSILENT simulations, highlighting both the analytical complexity and the practical implementation issues of load flow studies.

3. Collection of Background Information

3.1 Load Flow Analysis

Load flow studies form the backbone of system planning and operation, enabling engineers to determine voltage profiles, line flows, and system losses under steady-state conditions (Grainger & Stevenson, 2003). The Newton-Raphson method has emerged as the most widely used approach due to its quadratic convergence rate and high accuracy, especially compared with alternatives such as the Gauss-Seidel method (Glover et al., 2015). However, it is not without limitations — including high memory requirements for large Jacobian matrices and sensitivity to poor initial conditions (Stott & Alsac, 1974).

Top priority was given to examining the background Information in the context of the two test cases described in the problem statement:

- ***reactive power limit test***

In his study of the power flow problem, (McCalley, T7.0) points out to how generators are known to operate within defined limits, not only for their real power output (maximum and minimum) but also for their reactive power capabilities (maximum and minimum). The author goes on to state that the maximum reactive power capability represents the highest amount of reactive power a generator can produce while operating at a lagging power factor and that the minimum reactive power capability corresponds to the maximum reactive power the generator may absorb when operating with a leading power factor. As a supplement to the above-mentioned points, the author give reference to the *generator capability curve*, which indicates the maximum active and reactive power that can be delivered by a synchronous generator at its terminals (Enrique, 2014).

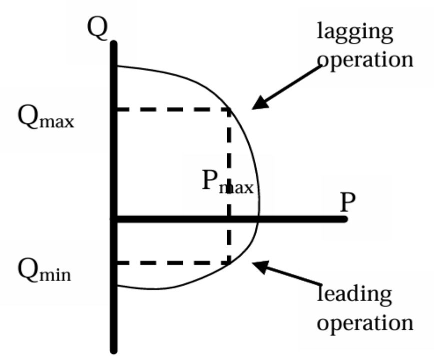


Figure 1: generator capability curve

In most power flow analyses, the reactive power limits of generators are explicitly modelled by assuming a somewhat restrained value for the maximum real power, P_{max} and then and then fixing the reactive limits Q_{min} (for the lagging limit) and Q_{max} (for the leading limit) as shown in the figure above.

It was on the basis of the above mentioned literature that in this study, the limits were approximated by the formula: Q_{min} = -0.5 × P and Q_{max} = +0.5 × P, where P is the rated real power output.

- ***Convergence test***

This test is crucial as it relates to determining the accuracy of the computed bus voltages. According to IEEE 3002.2-2018 standard which details out the "Recommended Practice for Conducting Load-Flow Studies and Analysis of Industrial and Commercial Power Systems", computed voltages are said to converge when the voltages get progressively closer to the actual solution with each iteration. The standard further points out that the convergence criterion defines accuracy requirements

According to the guidelines set forth in standard, the power mismatch check is a type of convergence criterion. The power mismatch tolerance is generally specified between 0.01 and 0.0001 per unit, referenced to the system MVA base.

In this study, the choice of convergence tolerance was guided by the principles established in this standard

3.2 Short-Circuit Studies

A short circuit current is the outcome of an undesired incident on a power system that needs to be managed without causing extensive damage (Sweeting, 2011). According to (IEC, 2016), there are, two types of short circuit current of different magnitudes are taken into account, specifically:

- The maximum short-circuits current, which specifies or governs the electrical equipment's capacity requirements. One example of such equipment is switchgears, and

- The minimum short-circuits current, serve as a reference point for tasks such as selecting appropriate fuses, configuring protective devices, and verifying motor start-up performance

There are different types of shorts circuit currents, namely: phase to phase, phase to neutral, phase to earth, and three phase short. Three phase short circuit fault is the most severe and destructive of them all and it is for these reasons that we are going to use this case, as it accounts for most extreme currents possible.

There are various standards used to calculate or compute the short circuit currents, such as Germany's VDE (Verband der Elektrotechnik) 0102 standard, ANSI (American National Standards Institute) C37.010 standard or the widely used International Electrotechnical Commission (IEC) 60909 standard. Our focus is specifically on the latter, which is IEC 60909. This standard is relevant for calculating short-circuit currents in low-voltage three-phase AC systems, and in high-voltage three-phase AC systems, operating at a nominal frequency of 50 Hz or 60 Hz (IEC, 2016).

According to (Sweeting, 2011), IEC 60909 standard outlines the methodology for calculating and measuring the short-circuit current waveform at a specific network location and time, either the point of initiation of the short circuit or contact opening of switchgear. Sweeting further outlines how the standard presents the concept of a voltage factor "c" to calculate the maximum and minimum short-circuit currents.

The voltage factor c is used to scale the equivalent voltage source in the calculations to compensate for changes in system voltage (EasyPower, 2025). Table 1 presented below outlines the criterion used in choosing the voltage factor.

Nominal voltage, U_n	Voltage factor c for calculation of	
	Maximum short-circuit currents (c_{max})[a]	Minimum short-circuit currents (c_{min})
Low voltage		
100–1000 V	1.05[b]	0.95[b]
(IEC 38, Table I)	1.10[c]	0.9[c]
High voltage[d]		
>1–35 kV	1.10	1.00
(IEC 38, Tables III and IV)		

Table 1: Voltage Factor c, according to IEEE 60909-0: 2016-10

The IEEE 4-Bus System has a base of 150 MVA and 132 kV, and as per the table above, the maximum and minimum voltage factors will be 1.10 and 1.00 respectively. In Power Factory, when configuring the short circuit current parameters, the c-factor values were set to the above-mentioned values, these are highlighted in blue in figures 18 and 19. Presented below are the results obtained from the short circuit study conducted in DIgSILENT:

3.3 Simulation Tools

• MATLAB (Simulink and M-file):
Offers flexible implementation of numerical algorithms such as Newton-Raphson, allowing detailed observation of intermediate steps. However, its reliance on user coding can introduce human error and requires robust debugging skills (Zimmerman et al., 2011).

• DIgSILENT PowerFactory:

A commercial-grade tool widely used in industry, providing built-in algorithms for load flow and fault studies with high reliability and visualization features. Its limitation lies in reduced transparency of the underlying iterative process, as it functions largely as a “black box” (DIgSILENT, 2020).

3.4 Advantages and Limitations of Newton-Raphson

• Advantages:
Fast convergence, accuracy, effectiveness in handling large-scale and meshed networks.

• Limitations:
Sensitive to initial guesses, computationally intensive for very large systems, potential difficulties near voltage collapse points (Kundur, 1994).

4. Procedure

Load flow study

The Newton–Raphson (NR) method is a numerical technique to solve nonlinear equations, and it is currently the most recognised technique for power flow analysis in electrical networks. The NR method linearizes the nonlinear power flow equations around an initial guess. It then updates bus voltage magnitudes and angles using an iterative procedure until power mismatches fall within a specified tolerance.

The process of iteratively updating line power injections and bus voltage angles relies on changes in active and reactive power (ΔP, ΔQ) associated with the corrections in voltage angles and magnitudes ($\Delta\delta$, ΔV). This men's, in part, the NR method's advantages of rapid quadratic convergence and its applicability to large, complicated systems.

In this project, the NR method offers an effective way to calculate load flow results and ensure the effective calculation of all bus voltages, power flows, and system balance, without relying solely on simulation.

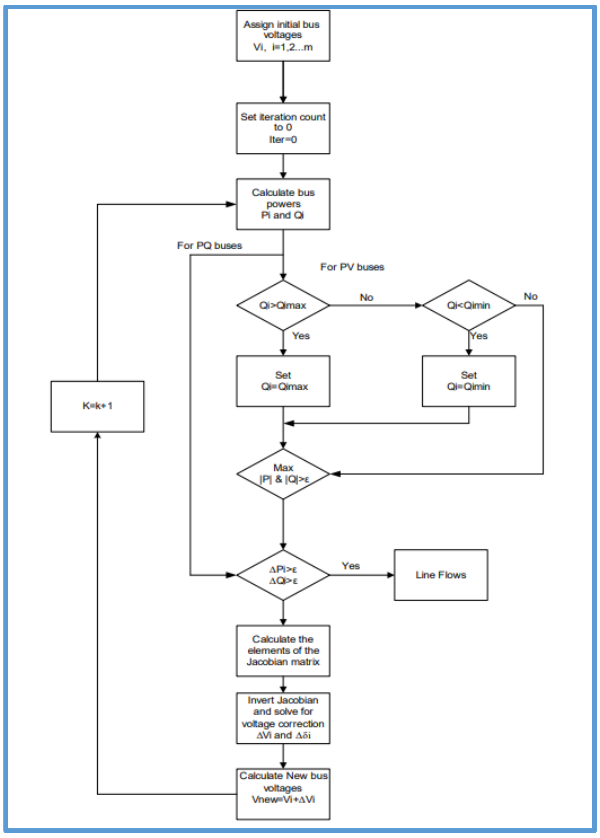


Figure 2: Procedure followed in conducting load flow

The flowchart illustrates the iterative procedure for solving the power flow problem using the Newton–Raphson method. The process begins with assigning initial bus voltages and setting the iteration counter. At each step, the active and reactive power at the buses are calculated, which form the basis for assessing the mismatches between the specified and calculated power values. For PQ buses, no additional checks are necessary, whereas for PV buses, special attention must be given to the reactive power limits in order to maintain physically feasible operation. If the calculated reactive power exceeds its maximum or falls below its minimum, it is fixed at the corresponding limit. This ensures that the generator operates within its reactive capability curve.

Following the power calculation, a convergence check is performed by evaluating the maximum of the absolute active and reactive power mismatches. If this maximum mismatch is smaller than a predefined threshold (epsilon), the iterative process is considered converged and the solution is accepted. Otherwise, the process proceeds to calculate the Jacobian matrix, which relates changes in voltage magnitudes and angles to changes in power mismatches. The Jacobian is then inverted or solved to obtain the necessary corrections for bus voltage magnitudes and phase angles.

Once the corrections are determined, the bus voltages are updated using the rule: the new value of each unknown variable equals its old value plus the computed correction. This step progressively refines the voltage profile until the mismatches fall below the tolerance. The iteration counter is then incremented, and the process repeats until convergence is achieved. The flowchart therefore provides a structured view of how the Newton–Raphson method ensures accuracy by combining power calculations, mismatch checks, Jacobian updates, and systematic voltage corrections in a looped manner until a stable operating point is found. Reference can be made to appendix A.

Short circuit study

1. Start

All specified parameters, notably fault impedance as per the brief, are documented and used to initialize other parameters such as the voltage factor.

2. Computing the total equivalent Impedances

As per the IEC 60909 standard, fault current is derived using the Thevenin equivalent from the fault point, which in this instance is Bus 1.

The total equivalent Impedance (Z_T) will be equal to the Thevenin equivalent as seen by the Bus 2 (algebraic sum of the generator, load, or lines that might be connected to the bus) and this is added to the fault impedance as given. In this case,

$$Z_T = Z_{bus} + Z_{fault}$$

3. Calculate short circuit currents

In accordance with the guidelines of (Sweeting, 2011), the upper-bound of the system operating voltage and lower-bound of the system operating voltage will be equal to $c_{max}U_n$ and $c_{min}U_n$ respectfully. Whereby $U_n$the nominal line to line voltage, which in is in this case is equal to the base voltage then by extension, using the formula as outlined in (Thurner & Braun, 2018):

$$I_k = \frac{cU_n}{\sqrt{3}\,|Z_T|}$$

Then other parameters dependent on I_k were calculated. Reference can be made to figures 28 and 29 as attached in appendix A.

5. Organizing Evidence

5.1 Load flow

5.1.1 MATLAB SCRIPT

Below are the MATLAB output images obtained from executing the script:

```
Y-Bus Matrix (Rectangular Form):
   (Real) + j(Imag)

   12.4740  -56.9323j      -3.4376  +18.2193j      -4.4459  +19.7006j      -4.5905  +19.0124j
   -3.4376  +18.2193j       7.4486  -36.8888j       0.0000   +0.0000j      -4.0110  +18.6695j
   -4.4459  +19.7006j       0.0000   +0.0000j       9.1695  -38.5574j      -4.7236  +18.8568j
   -4.5905  +19.0124j      -4.0110  +18.6695j      -4.7236  +18.8568j      13.3252  -56.5387j

Y-Bus Matrix (Polar Form - Magnitude ∠ Angle in degrees):

   58.2828 ∠  -77.64°      18.5408 ∠  100.68°      20.1960 ∠  102.72°      19.5587 ∠  103.57°
   18.5408 ∠  100.68°      37.6333 ∠  -78.58°       0.0000 ∠    0.00°      19.0955 ∠  102.13°
   20.1960 ∠  102.72°       0.0000 ∠    0.00°      39.6328 ∠  -76.62°      19.4394 ∠  104.06°
   19.5587 ∠  103.57°      19.0955 ∠  102.13°      19.4394 ∠  104.06°      58.0877 ∠  -76.74°
```

Figure 3: Matlab script results for the first iteration: Y-Bus matrices

The Y-Bus matrix shown in the figure above represents the admittance relationships between buses in the power system. In the rectangular form, each entry is expressed as a real part (conductance) plus an imaginary part (susceptance), while in the polar form it is expressed as a magnitude with an angle in degrees. Diagonal elements, which are large and positive, signify self-admittance (total admittance connected to a bus), whereas off-diagonal elements, usually negative, represent mutual admittance

Notice that some entries are exactly zero (for example, between Bus 2 and Bus 4 in the rectangular form). This indicates that there is no direct physical transmission line or branch connecting those two buses. In other words, power cannot flow directly between them, and any interaction must occur indirectly through other buses. Such zero elements are a normal feature of sparse Y-Bus matrices in real systems, since not all buses are interconnected.

The analysis then proceeded with the application of power flow equations

```
Bus 2: P_calc = -0.034376 pu, Q_calc = -0.182193 pu, ΔP = 2.010427e-01, ΔQ = 1.821932e-01
Bus 3: P_calc = -0.044459 pu, Q_calc = -0.197006 pu, ΔP = -4.822077e-01, ΔQ = 3.033949e-02
Bus 4: P_calc = -0.045905 pu, Q_calc = -0.190124 pu, ΔP = -5.207616e-01, ΔQ = -7.654314e-02
```

Figure 4: Matlab script results for the first iteration: power mismatches

The results above show the calculated real and reactive power at buses 2, 3, and 4, while the slack bus is excluded since it only serves as a reference for system voltage and angle. The calculated real and reactive power values represent the net injections at each bus, with negative signs indicating that these buses are primarily loads rather than generators. The corresponding mismatches in real and reactive power highlight the differences between the specified values and the computed ones.

Then before proceeding to computing the Jacobian matrices, the reactive power test at Bus 2.

```
Therefore conducting the Q-limit check on Bus 2:
 Q2 = -0.182193 pu, Qmin = -0.333333 pu, Qmax = 0.333333 pu
 Therefore, Q2 within the limits and as of such, the PV bus valid)
```

Figure 5: Matlab script results for the first iteration: reactive power test

For Bus 2, a reactive power limit check was carried out to ensure that the calculated reactive power output lies within the specified minimum and maximum bounds. The result shows that the reactive power value is within the defined range, meaning the generator is operating within its capability limits. As a result, Bus 2 can validly remain classified as a PV bus, since its reactive power requirement does not exceed the allowable limits

```
Jacobian Matrix [ΔP; ΔQ] = J × [Δδ; ΔV]:

State vector: x = [ δ2 δ3 δ4 |V3| |V4| ]
Mismatch vec: Δ = [ ΔP2 ΔP3 ΔP4 ΔQ3 ΔQ4 ]

Jacobian layout:

[ ∂ΔP2/∂δ2 ∂ΔP2/∂δ3 ∂ΔP2/∂δ4 ∂ΔP2/∂|V3| ∂ΔP2/∂|V4| ]
[ ∂ΔP3/∂δ2 ∂ΔP3/∂δ3 ∂ΔP3/∂δ4 ∂ΔP3/∂|V3| ∂ΔP3/∂|V4| ]
[ ∂ΔP4/∂δ2 ∂ΔP4/∂δ3 ∂ΔP4/∂δ4 ∂ΔP4/∂|V3| ∂ΔP4/∂|V4| ]
[ ∂ΔQ3/∂δ2 ∂ΔQ3/∂δ3 ∂ΔQ3/∂δ4 ∂ΔQ3/∂|V3| ∂ΔQ3/∂|V4| ]
[ ∂ΔQ4/∂δ2 ∂ΔQ4/∂δ3 ∂ΔQ4/∂δ4 ∂ΔQ4/∂|V3| ∂ΔQ4/∂|V4| ]

[Step C] NUMERICAL JACOBIAN (J):

   37.0710          0   -18.6695          0    -4.0110
         0    38.7544   -18.8568    -0.0889    -4.7236
  -18.6695   -18.8568    56.7288    -4.7236    -0.0918
         0    -9.2140     4.7236    77.3118    18.8568
    4.0110     4.7236   -13.3711    18.8568   113.2674
```

Figure 6: Matlab script results for the first iteration: Jacobian matrices

The Jacobian matrix shown here represents the sensitivity of real and reactive power mismatches with respect to bus voltage angles and magnitudes in the Newton–Raphson power flow method. Each element represents a partial derivative, measuring the impact of changes in angle or voltage at one bus on power at another. Zeros signify no direct influence.

```
[Step D] SOLUTION FOR THE STATE VECTORS:

 dδ2 = -0.00352253 rad, dδ3 = -0.02111518 rad, dδ4 = -0.01741593 rad
 d|V3| = -0.00066595 pu, d|V4| = -0.00161552 pu

[Step E] UPDATED STATE AFTER APPLYING dx:

 Bus 1: |V| = 1.010000 pu, angle = 0.000000 rad (0.000000°)
 Bus 2: |V| = 1.000000 pu, angle = -0.003523 rad (-0.201826°)
 Bus 3: |V| = 0.999334 pu, angle = -0.021115 rad (-1.209811°)
 Bus 4: |V| = 0.998384 pu, angle = -0.017416 rad (-0.997859°)

[Step F] VERIFICATION: RECOMPUTING INJECTIONS WITH UPDATING V

we must also alter our previous approximations for δ2, δ3, |V2|
Unknown Value_{new} = Unknown Value_{old} + Solved Δvalue

 Bus 2: P_new = 0.166647, Q_new = -0.193524, ΔP_new = 1.920892e-05, ΔQ_new = 1.935240e-01
 Bus 3: P_new = -0.531301, Q_new = -0.075352, ΔP_new = 4.634276e-03, ΔQ_new = -9.131469e-02
 Bus 4: P_new = -0.586292, Q_new = -0.144822, ΔP_new = 1.962553e-02, ΔQ_new = -1.218447e-01
```

Figure 7: Matlab script results for the first iteration: updated bus voltages

After applying the state vector corrections, the updated bus voltages show that Bus 1 maintains its reference value of 1.0100 pu at 0°, consistent with its role as the slack bus. The other buses experience slight adjustments in both magnitude and angle, reflecting the iterative refinement of the Newton–Raphson method. Specifically, Bus 2 remains at 1.0000 pu with a very small negative angle shift, while Bus 3 and Bus 4 settle at 0.9993 pu and 0.9984 pu, respectively, each with small negative angle deviations.

```
Therefore, in our case, the iterative process will be considered converged when ε is 1e-6:

max_i (|ΔP_i|, |ΔQ_i|) < 1e-06
```

Figure 8: Matlab script results for the first iteration: convergence test

Then from hereon, the solutions ought to be checked for convergence, if not, the second iteration is set in motion

```
[FINAL BUS VOLTAGES]
Bus 1: |V| = 1.010000 pu, angle = 0.000000 rad (0.000000°)
Bus 2: |V| = 1.000000 pu, angle = -0.003216 rad (-0.184284°)
Bus 3: |V| = 0.995493 pu, angle = -0.019925 rad (-1.141616°)
Bus 4: |V| = 0.995118 pu, angle = -0.016151 rad (-0.925371°)

[BUS INJECTIONS]
Bus 1: P = 0.932808 pu (139.921 MW), Q = 0.590222 pu (88.533 MVAr)
Bus 2: P = 0.166667 pu (25.000 MW), Q = -0.129851 pu (-19.478 MVAr)
Bus 3: P = -0.526667 pu (-79.000 MW), Q = -0.166666 pu (-25.000 MVAr)
Bus 4: P = -0.566667 pu (-85.000 MW), Q = -0.266666 pu (-40.000 MVAr)

[POWER BALANCE CHECK]
ΣP = 0.006141 pu (0.921 MW), ΣQ = 0.027039 pu (4.056 MVAr)
```

Figure 9: Matlab script results for the first iteration: final bus voltages

The final bus voltages show that all buses remain very close to 1 pu, which indicates good voltage stability across the system. Bus 1, the slack bus, is fixed at 1.01 pu with a 0° angle and serves as the system reference. Bus 2, a generator bus, maintains 1.0 pu with a small negative angle of -0.184°, reflecting power flow away from it. Buses 3 and 4, which are load buses, have voltages of 0.995 pu with slightly larger negative angles, but all values fall within the acceptable ±5% band. This suggests that the network is well regulated and no buses are suffering from voltage collapse or instability.

Looking at the bus injections, Bus 1 provides 139.9 MW and 88.5 MVAr, fulfilling its role as the slack bus that balances both real and reactive power. Bus 2 injects 25 MW but absorbs 19.5 MVAr, which may indicate that the generator is operating with limited reactive power capability. Bus 3 consumes 79 MW and 25 MVAr, while Bus 4 consumes 85 MW and 40 MVAr. Clearly, Buses 3 and 4 act as loads, drawing significant real and reactive power from the system, while Buses 1 and 2 cover this demand through generation.

The power balance check confirms that the network solution is accurate, with only a tiny mismatch of +0.921 MW and +4.056 MVAr across the entire system. These values are very small compared to the overall load and generation levels, and they simply reflect numerical tolerance from the Newton-Raphson iteration used in the power flow calculation. The convergence is therefore acceptable, and the results can be trusted for further analysis.

Finally, the signs of power injections provide insight into the operating condition of each bus. Positive active power means the bus is supplying real power, while negative values indicate demand. Similarly, positive reactive power means reactive support, and negative values mean absorption. In this case, Bus 1 supplies both P and Q, Bus 2 supplies P but absorbs Q, and Buses 3 and 4 absorb both P and Q. Overall, the generation of 165 MW matches the demand of 164 MW, and the system is balanced, with Bus 1 bearing the main responsibility for reactive support.

5.1.2 MATLAB SIMULINK

The screenshot attached below shows the IEEE 4 bus system designed in MATLAB Simulink:

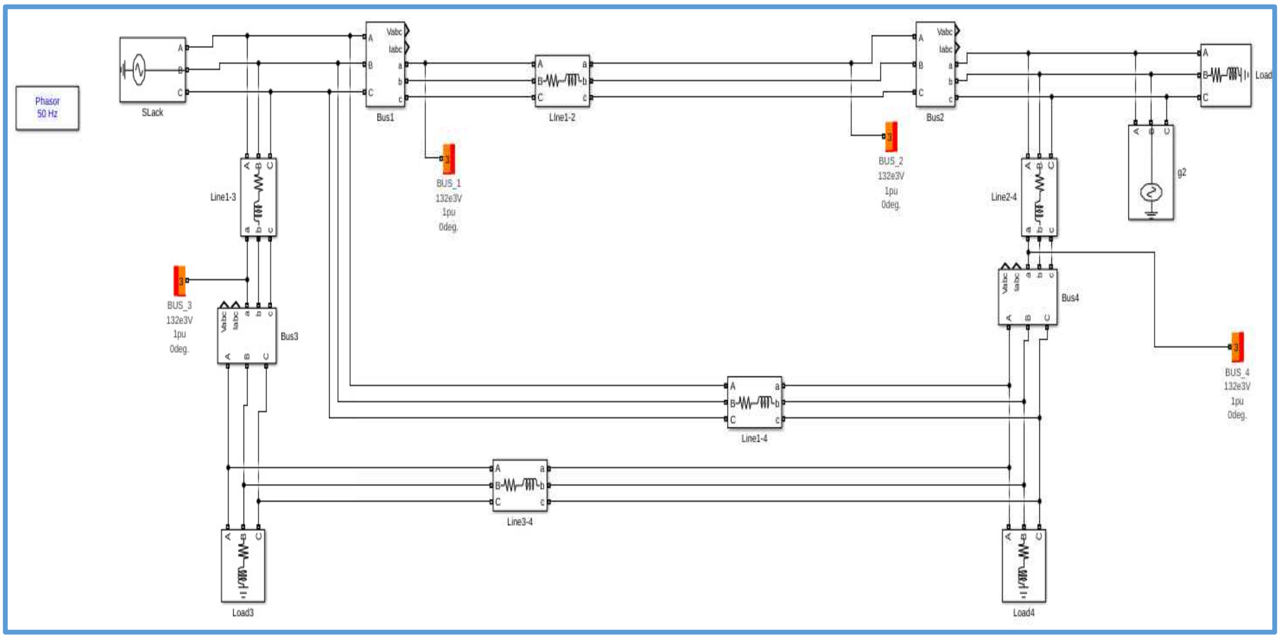


Figure 10: Matlab Simulink schematic IEEE 4 bus system

The figure shown above depicts results from the load flow study conducted in Simulink

Block name	Block type	Bus type	Bus ID	Vbase (kV)	Vref (pu)	Vangle (deg)	P (MW)	Q (Mvar)	Qmin (Mvar)	Qmax (Mvar)	V_LF (pu)	Vangle_LF (deg)	P_LF (MW)	Q_LF (Mvar)
SLack	Vsrc	swing	BUS_1	132.0000	1.0100	0	0.0100	0	-Inf	Inf	1.0100	0	139.9200	88.5104
Load2	RLC load	PQ	BUS_2	132.0000	1.0000	0	75.0000	50.0000	-Inf	Inf	1.0000	-0.1845	75.0000	50.0000
g2	Vsrc	PV	BUS_2	132.0000	1.0000	0	100.0000	0	-Inf	Inf	1.0000	-0.1845	100.0000	30.5422
Load3	RLC load	PQ	BUS_3	132.0000	1.0000	0	79.0000	25.0000	-Inf	Inf	0.9955	-1.1423	79.0000	25.0000
Load4	RLC load	PQ	BUS_4	132.0000	1.0000	0	85.0000	40.0000	-Inf	Inf	0.9951	-0.9259	85.0000	40.0000

Figure 11: Matlab Simulink results for short circuit study

The block highlighted in purple shows the final bus voltages and the block highlighted in green shows the final real and reactive power injected at the respective buses.

The yellow section represents the input system parameters defined for the load flow problem. These include the base voltage (132 kV), reference voltages in per unit, initial angle settings, and the specified active and reactive power values for each bus. These parameters form the foundation of the power flow model, as they describe the network operating conditions and bus specifications before running the numerical solution.

The results are consistent with those obtained in DIgSILENT and the MATLAB calculation, however, there is one difference that ought to be highlighted and this is highlighted in the blue block.

(McCalley, 1987) note that although generators can normally have either positive or negative reactive power injection. The sign convention for power injections depends on the operating mode: positive when a generator operates lagging and delivers reactive power to the bus, negative when it operates leading and absorbs reactive power, and zero at unity power factor. Loads typically have negative real and reactive power injections, indicating consumption, although in special modelling cases, they may be assigned positive real power injections.

Therefore, the net injected real and reactive power will be:

$$P_k = P_{gk} - P_{dk} \text{ and } Q_k = Q_{gk} - Q_{dk}$$

Where:

P_k= net real power injected
P_{gk}= real power generated
P_{dk}= real power demanded(load)

Q_k= net reactive power injected
Q_{gk}= reactive power generated
Q_{dk} = reactive power demanded(load)

$$P_k = 100\ MW - 75\ MW$$

$$P_k = 25\ MW$$

And

$$Q_k = 30.5422\ MVARS - 50\ MVARS$$

$$Q_k = -19.4578\ MVARS$$

Another important observation relates to the negative and positive signs attached to the power values. For bus 1, the real and reactive power are both positive, and this implies that the bus is supplying the real and reactive power, and this would be expected of the slack bus in this regard. For bus two, the reactive power is negative implying that the bus is absorbing reactive power.

5.1.3 DIgSILENT POWER FACTORY

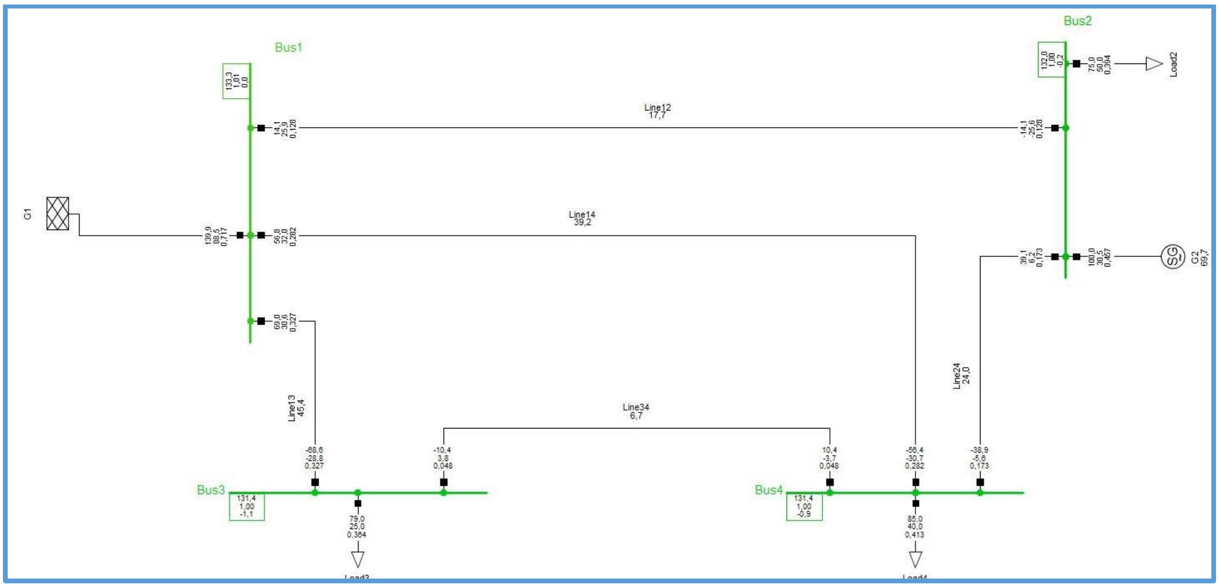


Figure 12: Power Factory DIgSILENT schematic for the IEEE 4 bus system

There is a strong positive correlation between the results obtained in MATLAB Simulink, M-File calculation and the results obtained in DIgSILENT PowerFactory. Just like MATLAB Simulink, in DIgSILENT PowerFactory the iterative AC load flow method for both balanced three-phase and unbalanced network conditions, employs an enhanced non-decoupled Newton-Raphson approach (DIgSILENT, 2025). By default, as in this study, the number iterations are set to three and the convergence tolerance (epsilon) is set to 10e-6. These are crucial as they have a direct impact on the accuracy of the solutions and even the computational time.

In their study, Afolabi et al. (2015) pointed out the following:

- Convergence tolerance (epsilon)

This is used to evaluate the precision of the solution. The higher the tolerance value, the higher the higher the accuracy and correspondingly, the lower the tolerance value, the less accurate.

- Number of iterations

As the bus count in the network increases, the iteration count for convergence also rises, but so does the accuracy. The number of iterations is directly proportional to the convergence tolerance used. The smaller the tolerance the greater number of iterations needed for the solution to converge.

- Computational time

In comparison to other methods such Gauss-Seidel and Fast decoupled, using the Newton-Raphson algorithm in DIgSILENT increases the computational (simulation) time. This is especially notable when the .M file is executed, in MATLAB (check the script), there is a 2 second delay before it could return the mismatch values.

All the above-mentioned factors lead to fundamental differences in the trailing digits of solutions between the three different simulation methods.

```
| Load Flow Calculation                                  Complete System Report: Substations, Voltage Profiles, Grid Interchange |
|    AC Load Flow, balanced, positive sequence                      |    Automatic Model Adaptation for Convergence             No    |
|    Automatic tap adjustment of transformers           No          |    Max. Acceptable Load Flow Error                                |
|    Consider reactive power limits                     No          |       Bus Equations (HV)                                1,00 kVA |
|                                                                   |       Model Equations                                   0,10 %   |
| Grid: Grid                  System Stage: Grid               | Study Case: Study Case                 | Annex:              / 1  |
|         rated                                Active  Reactive Power                         |                                     |
|        Voltage       Bus-voltage             Power   Power    Factor Current Loading        |         Additional Data             |
|         [kV]   [p.u.]     [kV]     [deg]      [MW]    [Mvar]   [-]    [kA]     [%]         |                                     |
|Bus1                                                                                          |                                     |
|          132,00      1,01   133,32     0,00                                                 |                                     |
|   Cub_5  /Xnet       G1                     139,92   88,51    0,85    0,72                  |Sk": 10000,00 MVA                    |
|   Cub_2  /Lne        Line12                  14,11   25,92    0,48    0,13    17,75         |Pv:      56,91 kW   cLod:  0,00 Mvar L:  100,00 km|
|   Cub_3  /Lne        Line14                  56,77   32,03    0,87    0,28    39,21         |Pv:     333,25 kW   cLod:  0,00 Mvar L:  100,00 km|
|   Cub_4  /Lne        Line13                  69,04   30,56    0,91    0,33    45,41         |Pv:     406,04 kW   cLod:  0,00 Mvar L:  100,00 km|
|Bus2                                                                                          |                                     |
|          132,00      1,00   132,00    -0,18                                                 |                                     |
|   Cub_2  /Sym        G2                     100,00   30,54    0,96    0,46    69,71         |Typ:    PQ                           |
|   Cub_3  /Lod        Load2                   75,00   50,00    0,83    0,39                  |Pl0:     75,00 MW   Ql0:  50,00 Mvar  |
|   Cub_1  /Lne        Line12                 -14,05  -25,62   -0,48    0,13    17,75         |Pv:      56,91 kW   cLod:  0,00 Mvar L:  100,00 km|
|   Cub_4  /Lne        Line24                  39,05    6,16    0,99    0,17    24,02         |Pv:     114,64 kW   cLod:  0,00 Mvar L:  100,00 km|
|Bus3                                                                                          |                                     |
|          132,00      1,00   131,40    -1,14                                                 |                                     |
|   Cub_2  /Lod        Load3                   79,00   25,00    0,95    0,36                  |Pl0:     79,00 MW   Ql0:  25,00 Mvar  |
|   Cub_1  /Lne        Line13                 -68,64  -28,76   -0,92    0,33    45,41         |Pv:     406,04 kW   cLod:  0,00 Mvar L:  100,00 km|
|   Cub_3  /Lne        Line34                 -10,36    3,76   -0,94    0,05     6,73         |Pv:      10,22 kW   cLod:  0,00 Mvar L:  100,00 km|
```

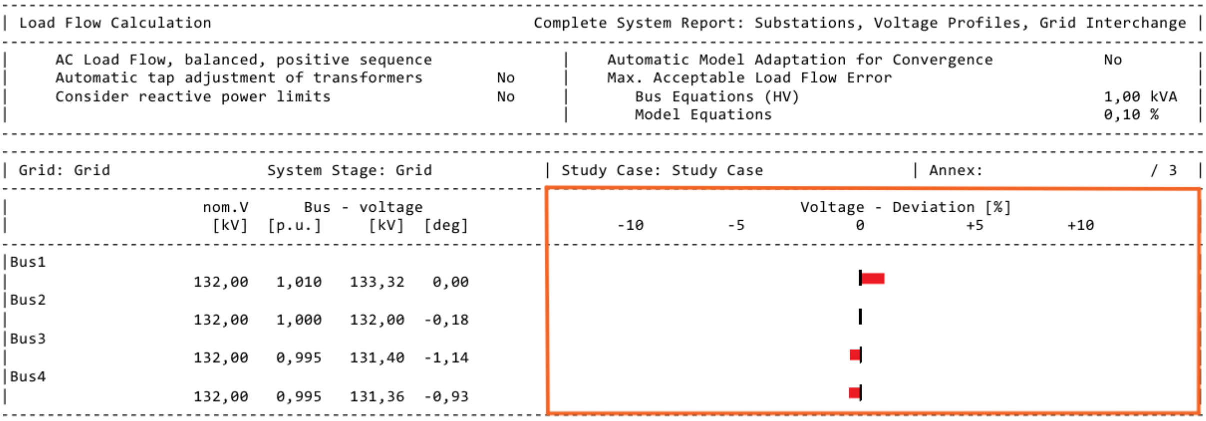
Figure 13: DIgSILENT load flow results: bus 1-3

```
| Grid: Grid                  System Stage: Grid               | Study Case: Study Case                 | Annex:              / 2  |
|         rated                                Active  Reactive Power                         |                                     |
|        Voltage       Bus-voltage             Power   Power    Factor Current Loading        |         Additional Data             |
|         [kV]   [p.u.]     [kV]     [deg]      [MW]    [Mvar]   [-]    [kA]     [%]         |                                     |
|Bus4                                                                                          |                                     |
|          132,00      1,00   131,36    -0,93                                                 |                                     |
|   Cub_2  /Lod        Load4                   85,00   40,00    0,90    0,41                  |Pl0:     85,00 MW   Ql0:  40,00 Mvar  |
|   Cub_1  /Lne        Line14                 -56,44  -30,65   -0,88    0,28    39,21         |Pv:     333,25 kW   cLod:  0,00 Mvar L:  100,00 km|
|   Cub_3  /Lne        Line24                 -38,94   -5,62   -0,99    0,17    24,02         |Pv:     114,64 kW   cLod:  0,00 Mvar L:  100,00 km|
|   Cub_4  /Lne        Line34                  10,38   -3,72    0,94    0,05     6,73         |Pv:      10,22 kW   cLod:  0,00 Mvar L:  100,00 km|
```

Figure 14: DIgSILENT load flow results: bus 4

The highlighted results show the final bus voltages, active power, and reactive power across the four buses in the system. Bus 1, acting as the slack bus, maintains a slightly elevated voltage of 1.01 pu (133.32 kV) at 0°, supplying both active (139.92 MW) and reactive (88.51 Mvar) power to balance system losses and meet demand. The remaining buses—Bus 2, Bus 3, and Bus 4—are operating close to nominal voltage (around 1.00 pu), with minor deviations in angle (ranging from –0.18° to –1.14°). This confirms that the voltage profile is well-regulated across the network, staying close to rated values while still reflecting the effects of loading.

From the power injections, we observe that Bus 2 (100 MW, 30.54 Mvar), Bus 3 (79 MW, 25 Mvar), and Bus 4 (85 MW, 40 Mvar) are primarily load buses, with active power largely consumed and smaller amounts of reactive support drawn. The power factor values (ranging from 0.85 to 0.96) indicate that the loads are slightly inductive but within acceptable operating limits. Overall, the results demonstrate a stable operating condition where voltages remain close to nominal, the slack bus successfully balances system mismatches, and reactive power demands are within expected ranges. This suggests the system is operating efficiently without significant risk of voltage instability.



```
| Load Flow Calculation                                   Complete System Report: Substations, Voltage Profiles, Grid Interchange |
|     AC Load Flow, balanced, positive sequence                 |     Automatic Model Adaptation for Convergence         No     |
|     Automatic tap adjustment of transformers        No        |     Max. Acceptable Load Flow Error                           |
|     Consider reactive power limits                  No        |        Bus Equations (HV)                            1,00 kVA |
|                                                               |        Model Equations                               0,10 %   |
| Grid: Grid                  System Stage: Grid          | Study Case: Study Case                  | Annex:              / 3 |
|                    nom.V      Bus - voltage                               Voltage - Deviation [%]
|                     [kV]   [p.u.]      [kV]   [deg]          -10         -5          0          +5         +10
|Bus1
|                   132,00   1,010    133,32    0,00
|Bus2
|                   132,00   1,000    132,00   -0,18
|Bus3
|                   132,00   0,995    131,40   -1,14
|Bus4
|                   132,00   0,995    131,36   -0,93
```

Figure 15: DIgSILENT load flow results: voltage deviation

The load flow results indicate that all bus voltages remain within an acceptable deviation range from the nominal 132 kV. Bus 1 shows a slight overvoltage at 1.010 p.u. (133.32 kV), while Buses 2, 3, and 4 exhibit marginals under voltages of 1.000 p.u. (132.00 kV), 0.995 p.u. (131.40 kV), and 0.995 p.u. (131.36 kV), respectively.

Voltage deviations (+1.0% to -0.5%) are within the ±5% limits, attributed to line impedance and load demand. The system's voltage profile shows stable operation, meeting standard tolerance criteria.

Grid	DIgSILENT		MATLAB calculation		MATLAB Simulink	
	|V|	θ	|V|	θ	|V|	θ
Bus 1	1.010	0.00	1.010000	0.000000	1.0100	0.0000
Bus 2	1.000	-0.18	1.000000	-0.184284	1.0000	-0.1845
Bus 3	0.995	-1.14	0.995493	-1.141616	0.9955	-1.1423
Bus 4	0.995	-0.93	0.995118	0.925371	0.9951	-0.9259

Table 2: Table showing the bus voltages from three different simulation programmes

The table compares bus voltage magnitudes and angles obtained from DIgSILENT, MATLAB numerical calculation, and MATLAB Simulink. Overall, the results show strong correlation across the three methods, with only slight deviations observed in the voltage magnitudes (to the fourth decimal place) and in the angles (to the third decimal place). These minor differences can be attributed to factors such as the number of iterations performed during the solution process, the convergence tolerance specified, and inherent simulation error percentages within each platform. Despite these variations, the results remain highly consistent and accurate, confirming that all three approaches provide reliable solutions for the load flow study

5.2 Short circuit current

5.2.1 Matlab Script

The Figure shown below shows the MATLAB script short circuit fault current results

```
INPUTS
  Target bus: 1
  Un (kV): 132.000    |   c_max: 1.10   c_min: 1.00
  Zf (ohm): 20.000+40.000j    |   Ssc at slack (MVA): 7200.0   |   R_scale_min: 1.20
  Zs at slack (ohm): 2.420000

THEVENIN EQUIVALENTS (at Bus 1)
  Z_th MAX:  2.420000 -0.000000j  ohm
  Z_th MIN:  2.420000 -0.000000j  ohm
  Z_total incl. Zf (MAX):  22.420000 +40.000000j  ohm
  Z_total incl. Zf (MIN):  22.420000 +40.000000j  ohm

RESULTS (BUS 1 ONLY)
+---------------------------+------------+--------------+-------------+
| Quantity                  |  Unit      |  MAX case    |  MIN case   |
+---------------------------+------------+--------------+-------------+
| Ik" (initial 3phi)        |  kA        |       1.828  |       1.662 |
| Sk" (short-circ power)    |  MVA       |     417.981  |     379.983 |
+---------------------------+------------+--------------+-------------+

R/X at Bus 1 (MAX-case Z_th): 95172262285051.7031
```

Figure 16: Matlab script for the short circuit study

From the figure shown above, the minimum and maximum short circuits are 1.662 kA and 1.824 kA respectively.

5.2.2 DIgSILENT

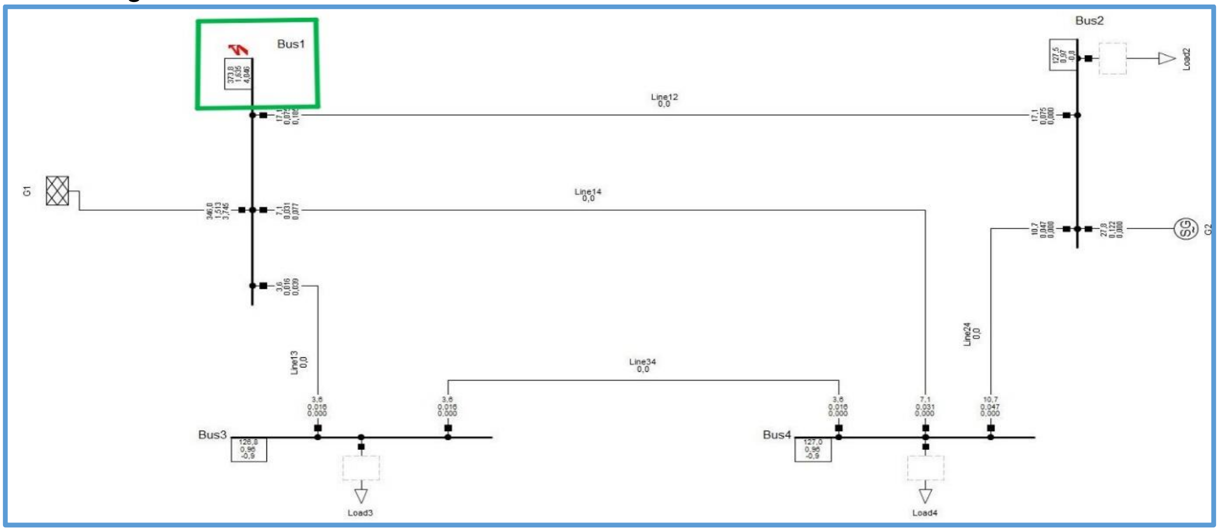


Figure 17: IEEE 4 bus system depicted for during the short circuit study in DIgSILENT

The green line demarcated area shows the results obtained during the short circuit study

```
info - Short-circuit calculation started...
info - Short-circuit calculated at Terminal Grid\Bus1.ElmTerm
info - Short-circuit calculation successfully executed.
-------------------------------------------------------------------------------------------------------
|                 |                                                 |   DIgSILENT   | Project:          |
|                 |                                                 | PowerFactory  |-------------------|
|                 |                                                 |   2025 SP1    | Date:  2025/08/20 |
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
| Fault Locations with Feeders                                                                        |
| Short-Circuit Calculation / Method : IEC 60909               3-Phase Short-Circuit   / Min. Short-Circuit Currents |
-------------------------------------------------------------------------------------------------------
| Asynchronous Motors                      | Grid Identification              | Short-Circuit Duration        |
|    Always Considered                     |    Automatic                     |    Break Time           0,10 s |
|                                          |                                  |    Fault Clearing Time (Ith) 1,00 s |
| Decaying Aperiodic Component (idc)       | Conductor Temperature            | Voltage factor c              |
|    Using Method                B         |    User Defined           No     |    User defined equivalent voltage source fact
|                                          |                                  |
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
| Grid: Grid                  System Stage: Grid         |                          | Annex:        / 1  |
-------------------------------------------------------------------------------------------------------
|                    rtd.V.    Voltage      c-       Sk"          Ik"            ip      Ib     Sb      Ik     Ith |
|                     [kV]   [kV]  [deg]  Factor   [MVA]      [kA]    [deg]     [kA]    [kA]   [MVA]   [kA]   [kA] |
|                                                                                                       |
|Bus1               132,00  126,65  -0,89  1,00   373,83 MVA  1,64 kA  -64,32   4,05 kA  1,64  373,83  1,64   1,66 |
|    Line12          Bus2                          17,14 MVA  0,07 kA  113,60   0,19 kA                         |
|    Line14          Bus4                           7,07 MVA  0,03 kA  116,05   0,08 kA                         |
|    Line13          Bus3                           3,61 MVA  0,02 kA  115,97   0,04 kA                         |
|    G1                                           346,02 MVA  1,51 kA  -64,23   3,74 kA                         |
-------------------------------------------------------------------------------------------------------
```

Figure 18: Results for the minimum short circuit study in DIgSILENT

In accordance with IEC 60909 approach, the highlighted red column presents five parameters that are used to define all short-circuit tests and the calculation of the potential short-circuit currents in power systems. Ik" is the initial symmetrical short-circuit current, Ip refers to the Peak short-circuit current, Ib is the symmetrical short-circuit breaking current, Ik steady-state short-circuit current and Ith is the thermal-equivalent short-circuit current. The most significant of these is the initial symmetrical short-circuit current (Ik") as it is primarily used to determine the equipment rating (Bandaru et al., 2022).

From the minimum short circuit current study, the parameter that holds utmost significance is Ik". (Schneider Electric, 2025) states that in selecting relays, a reasonable margin should be. Maintained between the overcurrent settings and the fault current level, with settings being lower. In this context, the minimum Ik" value represents the lowest fault current that must be sensed by the protection system. When the overcurrent protection elements are set beneath this value, the system should be able to detect faults, even amidst worst-case conditions.

```
info - Short-circuit calculation started...
info - Short-circuit calculated at Terminal Grid\Bus1.ElmTerm
info - Short-circuit calculation successfully executed.
------------------------------------------------------------------------------------------------------
|               |                                                  |   DIgSILENT   | Project:                  |
|               |                                                  |  PowerFactory |----------------------------
|               |                                                  |   2025 SP1    | Date:  2025/08/20         |
------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------
| Fault Locations with Feeders                                                                         |
| Short-Circuit Calculation / Method : IEC 60909              3-Phase Short-Circuit    / Max. Short-Circuit Currents |
------------------------------------------------------------------------------------------------------
| Asynchronous Motors                | Grid Identification                 | Short-Circuit Duration                  |
|    Always Considered               |    Automatic                        |    Break Time                   0,10 s  |
|                                    |                                     |    Fault Clearing Time (Ith)    1,00 s  |
| Decaying Aperiodic Component (idc) | Conductor Temperature               | Voltage factor c                        |
|    Using Method              B     |    User Defined             No      |    User defined equivalent voltage source fact
|                                    |                                     |    Equivalent voltage source factor  1,10 |
------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------
| Grid: Grid              System Stage: Grid         |                            | Annex:          / 1      |

                   rtd.V.    Voltage      c-      Sk"          Ik"           ip      Ib      Sb      Ik     Ith
                    [kV]    [kV]  [deg] Factor   [MVA]      [kA]    [deg]    [kA]    [kA]   [MVA]    [kA]   [kA]

|                                                                                                            |
|Bus1              132,00  139,96  -0,79   1,10  413,11 MVA  1,81 kA  -64,22   4,47 kA  1,81  413,11   1,81   1,84 |
|    Line12          Bus2                         17,05 MVA  0,07 kA  113,50   0,18 kA                            |
|    Line14          Bus4                          7,06 MVA  0,03 kA  115,53   0,08 kA                            |
|    Line13          Bus3                          3,60 MVA  0,02 kA  115,46   0,04 kA                            |
|    G1                                          385,41 MVA  1,69 kA  -64,12   4,17 kA                            |
------------------------------------------------------------------------------------------------------
```

Figure 19: Results for the maximum short circuit study in DIgSILENT

To better understand the parameters obtained from the minimum short circuit current study, reference is made to (Sweeting, 2011), whereby it is stated that the rated short-circuit current determines three performance criteria for a piece of equipment, each of which must be verified through testing:

1. Irms (rated) > Irms (calculated) = Ik.

This means that the rated root mean square (RMS) initial symmetrical short-circuit current of the bus bar must exceed the calculated RMS short-circuit current. In this case, the bus bar's rating must be higher than the initial symmetrical short-circuit current of 1.64 kA. And similarly:

2. Ip(rated) > Ip(calculated) = Ip

The rated peak withstand current must be greater than the calculated peak short-circuit value.

3. $I_{rms}^2 T_{rated}$> $I_{rms}^2 T$ (calculated) =$I_k^2 T$

The rated short-time withstand current, must be greater than the calculated thermal stress, thus ensuring protection against the electromagnetic forces at the bus bar.

Conducting all three tests is mandatory for preventing equipment failure.

From the table shown above, it is can be observed that the results obtained are very similar with only a 0.01 kA difference, confirming that the fault current is indeed correct.

It is similarly important to make mention of the short circuit duration time parameters.

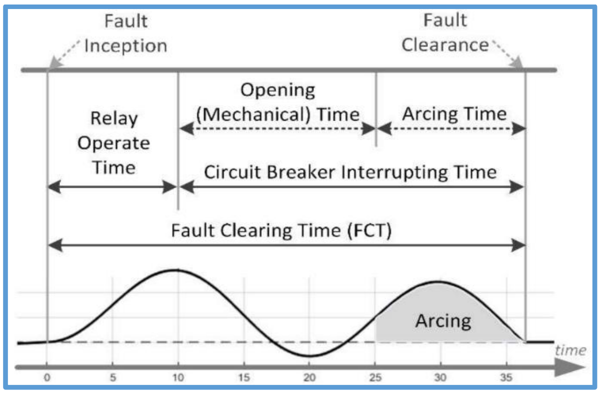


Figure 20: fault clearing time curve

According to (Zubić et al, 2021), the *fault clearing time* is the duration between fault occurrence and the moment when fault current is interrupted. The Fault clearing time is made up of the relay operating time (time from fault identification to relay trip signal transmission) and the *circuit breaker interrupting time* (time from trip signal reception to actual circuit breaker's physical disconnection of fault current) check figure above:

From both results in figures 1 & 2, the default fault clearing time is 1 second, meaning the relay operating time and circuit breaker interrupting time is 0.90 and 0.10 seconds respectively. This means that if the circuit breaker detects a minimum short circuit of 1.64 kA, then within 1 second the system should clear.

short circuit current type	DIgSILENT PowerFactory	matlab m-file calculation
min. short circuit current	1.64 kA	1.662 kA
max. short circuit current	1.81 kA	1.828 kA

Table 3: Comparison between the minimum and maximum short circuit study from DIgSILENT and .m file

From the table shown above, it is can be observed that the results obtained are very similar with only marginal difference, confirming that the fault current is indeed correct.

6. Conclusions and recommendations

Conclusion

In conclusion, this study successfully performed load flow and short-circuit analysis of the IEEE 4-bus power system using MATLAB (Simulink and M-file scripting) and DIgSILENT PowerFactory. The Newton-Raphson method converged reliably, and the solution confirmed that Bus 2 operated within its reactive power limits, validating its status as a PV bus. Results from MATLAB coding, Simulink models, and DIgSILENT showed close agreement with manual calculations, confirming the correctness of both the modelling and iterative procedure.

The IEC 60909 short-circuit study produced consistent minimum fault current values, demonstrating how Thevenin equivalents and fault impedances influence system fault levels. These findings enhance understanding of power system principles and demonstrate the value of comparing analytical and simulation approaches for validation.

Recommendations
For future studies, it is recommended that:

1. Systematic comparison of M-file, Simulink, and PowerFactory results continues, to confirm consistency and reveal modelling discrepancies.

2. In short-circuit analysis, investigate different c-factor values and unbalanced fault types to extend IEC 60909 application. Similarly, there should be more emphasis on both the minimum and maximum short circuit current

7. Challenges Encountered and Resolutions

7.1 Challenges

- In conducting the literature review for the short circuit study, it challenging because access to the IEC 60909 standard is either subscription-based or requires purchase.
- In the initial phase of the project, access to the PowerFactory software was restricted, and as of such, progress was severely hampered.
- As much as the Newton Raphson iteration is said to be faster, easier, accurate and effective but it was challenging to construct both Y-bus matrix and Jacobian matrix and doing multiple number of iterations (maximum of 40 iterations) to prevent convergence problems.

7.2 Resolutions

- Used existing literature as prescribed in the brief and supplemented with previews of the standard.
- Resorted to installing the software on a personal computer, to allow for easier and timely access.
- Comparison of manual calculations against Matlab Simulink/DIgSILENT simulation results to mitigate any potential errors beforehand.

8. References

• Anderson, P. (1995). Analysis of Faulted Power Systems. Wiley-IEEE Press.

• CIGRE (2014). Short-Circuit Currents in Power Systems in Accordance with IEC 60909. CIGRE Technical Brochure.

• DIgSILENT (2020). PowerFactory User Manual. Gomaringen: DIgSILENT GmbH.

• ECSA (2019). Graduate Attributes and Competencies. Engineering Council of South Africa.

• Glover, J. D., Sarma, M. S. & Overbye, T. (2015). Power System Analysis and Design. 6th ed. Cengage Learning.

• Grainger, J. J. & Stevenson, W. D. (2003). Power System Analysis. McGraw-Hill.

• IEC (2016) IEC 60909-0:2016. Short-circuit currents in three-phase a.c. systems - Part 0: Calculation of currents. 2nd ed. International Electrotechnical Commission.

•"What Is a Short Circuit and Why Does It Happen?" [Webpage]. Published May 7, 2025. Available at: (https://www.psolera.com/en/news/what-is-a-short-circuit-and-why-it-happens) [Accessed: 19 August 2025].

• IEC 60909 Short-Circuit in EasyPower [Webpage]. Available at: https://www.easypower.com/resources/article/iec-60909-short-circuit-in-easypower [Accessed: 19 August 2025]

• Bandaru, D., Sura, S.R., Bokam, J.K. and Kumar, K.A., 2022. Methods for short circuit analysis in ANSI and IEC standards. International Journal of Mechanical Engineering, 7(2), p.2643.

• Kundur, P. (1994). Power System Stability and Control. McGraw-Hill.

• Pai, M. A. (2012). Computer Techniques in Power System Analysis. McGraw-Hill.

• Stott, B. & Alsac, O. (1974). Fast Decoupled Load Flow. IEEE Transactions on Power Apparatus and Systems, 93(3), pp. 859–869.

• Zimmerman, R. D., Murillo-Sánchez, C. E. & Thomas, R. J. (2011). MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education. IEEE Transactions on Power Systems, 26(1), pp. 12–19

• Enrique, E.H., 2014, July. Generation capability curves for wind farms. In 2014 IEEE Conference on Technologies for Sustainability (SusTech) (pp. 103-106). IEEE

• Sweeting, D., 2011. Applying IEC 60909, fault current calculations. IEEE Transactions on Industry Applications, 48(2), pp.575-580.

• Schneider Electric (2025). C1 Overcurrent Protection for Phase and Earthfaults [PDF]. Available at: (https://www.se.com/ww/en/tools/npag-online-re14y/pdf/C1-Overcurrent_Protection_for_Phase_and_Earthfaults.pdf) [Accessed: 20 August 2025].

• Zubić, S., Gajić, Z., Naidu, O.D. and Kockott, M., line protection operate time: speed vs. circuit breaker wear, power system stability and security.

• DIgSILENT. (2023). Load Flow Analysis [Online]. Available from: (Load Flow Analysis - DIgSILENT) [Accessed 22 August 2023]

• Afolabi, O.A., Ali, W.H., Cofie, P., Fuller, J., Obiomon, P. and Kolawole, E.S., 2015. Analysis of the load flow problem in power system planning studies. Energy and Power Engineering, 7(10), pp.509-523.

Thurner, L. and Braun, M., 2018, October. Vectorized calculation of short circuit currents considering distributed generation-an open source implementation of iec 60909. In 2018 IEEE PES innovative smart grid technologies conference europe (ISGT-Europe) (pp. 1-6). IEEE.

9. Appendices

Appendix A: MATLAB M-File Code for Newton-Raphson Load Flow

```matlab
fprintf('================================================================================\n');


%% -----------------------------------------------------------------------------------------
%% -------------------------
%% Line Impedance Data: [From, To, R, X]
%% -------------------------
line_data = [
    1 2 0.0100 0.0530;
    1 3 0.0109 0.0483;
    1 4 0.0120 0.0497;
    2 4 0.0110 0.0512;
    3 4 0.0125 0.0499
];

% Convert to MATLAB table and display it
LineTable = array2table(line_data, 'VariableNames', {'From','To','R','X'});
disp('Line Impedance Data (Ohmic):');
disp(LineTable);

%% -----------------------------------------------------------------------------------------
%% (Do NOT clc here, or you will erase the table output)
%% -----------------------------------------------------------------------------------------


%% ------------------------------------------------------INTRODUCTION SECTION----------------------

fprintf('\nWe classify buses in a power system to determine what quantities are known and what we\n');
fprintf('need to solve for during a load flow (power flow) analysis.\n\n');

% Print the bus/type table (ASCII borders)
fprintf('+-----+---------------+------------------+------------------+\n');
fprintf('| %-3s | %-13s | %-18s | %-18s |\n', 'bus', 'type', 'known variables', 'unknown variables');
fprintf('+-----+---------------+------------------+------------------+\n');
fprintf('| %-3d | %-13s | %-18s | %-18s |\n', 1, 'Slack', '-', '-');
fprintf('| %-3d | %-13s | %-18s | %-18s |\n', 2, 'PV (generatr)', 'P₂, V₂', 'δ₂, Q₂');
fprintf('| %-3d | %-13s | %-18s | %-18s |\n', 3, 'PQ (load)',     'P₃, Q₃', 'δ₃, V₃');
fprintf('| %-3d | %-13s | %-18s | %-18s |\n', 4, 'PQ (load)',     'P₄, Q₄', 'δ₄, V₄');
fprintf('+-----+---------------+------------------+------------------+\n\n');

% Notes
fprintf('• For a PQ bus, we need to calculate both voltage magnitude and angle.\n\n');
fprintf('• For a PV bus, we calculate only the angle and reactive power.\n\n');
fprintf('• The slack bus serves as a phase reference and absorbs the difference\n');
fprintf('  in generation/load mismatches.\n\n');

%% ----------------------------------------------------END INTRODUCTION SECTION--------------------
```

```matlab
%% --------------------------------------------------CONTRUCTING THE Y-BUS MATRIX----------------
n_bus = 4;
Ybus  = zeros(n_bus);

% -------------------------
% Construct Y-Bus
% -------------------------

fprintf(' FORMING THE Y-BUS MATRIX :\n\n');

fprintf('Using the inspection method:\n\n');

fprintf('  (1) For the self-admittance or diagonal elements:\n\n');
fprintf('      Yii = Σ (Yij)   at bus i\n');
fprintf('              j=1..n\n\n');

fprintf('  (2) For the mutual-admittance or off-diagonal elements:\n\n');
fprintf('      Yij = -(line admittance between i and j)\n');


for k = 1:size(line_data,1)
    i = line_data(k,1);
    j = line_data(k,2);
    R = line_data(k,3);
    X = line_data(k,4);
    Z = complex(R, X);
    y = 1 / Z;
    Ybus(i,j) = Ybus(i,j) - y;
    Ybus(j,i) = Ybus(j,i) - y;
    Ybus(i,i) = Ybus(i,i) + y;
    Ybus(j,j) = Ybus(j,j) + y;
end

% -------------------------
% Display in Rectangular Form
% -------------------------
fprintf('\nY-Bus Matrix (Rectangular Form):\n');
fprintf('   (Real) + j(Imag)\n\n');
for i = 1:n_bus
    for j = 1:n_bus
        fprintf('%10.4f%+10.4fj\t', real(Ybus(i,j)), imag(Ybus(i,j)));
    end
    fprintf('\n');
end

% -------------------------
% Display in Polar Form
% -------------------------
fprintf('\nY-Bus Matrix (Polar Form - Magnitude ∠ Angle in degrees):\n\n');
for i = 1:n_bus
    for j = 1:n_bus
        fprintf('%10.4f ∠ %7.2f°\t', abs(Ybus(i,j)), angle(Ybus(i,j))*180/pi);
    end
    fprintf('\n');
end

%% -------------------------------------------------- END CONTRUCTING THE Y-BUS MATRIX----------
```

Figure 21: matlab code showing the introduction section (left) and formation of the y-bus matrix (right)

```matlab
%% ------------------------------------------------ SPECITYING THE SYSTEM RELATED PARAMETERS---------------

%% Assumed Flat Start Voltages
V = [1.01+0j, 1.00+0j, 1.00+0j, 1.00+0j];

%% Specified Power (P_spec and Q_spec) based on S_base = 150 MVA
S_base = 150;
P_gen_MW   = [0, 100, 0, 0];
Q_gen_MVar = [0,  50, 0, 0];
P_load_MW   = [0, 75, 79, 85];
Q_load_MVar = [0, 50, 25, 40];

P_spec = (P_gen_MW   - P_load_MW)   / S_base;
Q_spec = (Q_gen_MVar - Q_load_MVar) / S_base;
P_spec(1) = NaN;  Q_spec(1) = NaN;   % Slack bus

% PV bus 2 reactive limits (±0.5*P rating in pu)
Qmax_pu = +0.5 * (P_gen_MW(2) / S_base);
Qmin_pu = -0.5 * (P_gen_MW(2) / S_base);

%% INITIAL MISMATCH (single evaluation before iterations)
DeltaP = zeros(n_bus,1);
DeltaQ = zeros(n_bus,1);
for bus = 2:n_bus
    I = Ybus(bus,:) * V.';
    S = V(bus) * conj(I);
    DeltaP(bus) = P_spec(bus) - real(S);
    DeltaQ(bus) = Q_spec(bus) - imag(S);
end

%% Convergence parameters
eps_tol   = 1e-6;
max_iter  = 100;
converged = false;

% Keep history if desired (not printed)
iter_history = struct('iter',{}, 'maxMismatch',{}, 'V',{}, 'DeltaP',{}, 'DeltaQ',{});

% Verbosity control
show_full_detail_first_iter = true;

%% ------------------------------------------------END SPECITYING THE SYSTEM RELATED PARAMETERS-----------
```

```matlab
%%                                  ITERATIVE NEWTON-RAPHSON LOOP
%% ----------------------------------------------------------------------------------------------------------
for iter = 1:max_iter

   %% --- Step A: Compute current mismatches (ΔP, ΔQ) with current V


   DeltaP = zeros(n_bus,1);
   DeltaQ = zeros(n_bus,1);
   P_calc = zeros(n_bus,1);
   Q_calc = zeros(n_bus,1);
   for bus = 2:n_bus
      I = Ybus(bus,:) * V.';
      S = V(bus) * conj(I);
      P_calc(bus) = real(S);
      Q_calc(bus) = imag(S);
      DeltaP(bus) = P_spec(bus) - P_calc(bus);
      DeltaQ(bus) = Q_spec(bus) - Q_calc(bus);
   end

   % Build mismatch vector in ordering [ΔP2 ΔP3 ΔP4 ΔQ3 ΔQ4]
   mism_vec    = [DeltaP(2); DeltaP(3); DeltaP(4); DeltaQ(3); DeltaQ(4)];
   maxMismatch = max(abs(mism_vec));

   % Save (not printed)
   iter_history(end+1).iter        = iter;
   iter_history(end).maxMismatch   = maxMismatch;
   iter_history(end).V             = V;
   iter_history(end).DeltaP        = DeltaP;
   iter_history(end).DeltaQ        = DeltaQ;

   %% Print control
   if iter == 1 && show_full_detail_first_iter
      fprintf('\n================================== Iteration %d (FULL DETAIL) ==================================\n\n', iter);

      fprintf('[STEP A] INJECTED POWERS & INITIAL MISMATCHES:\n\n');


fprintf('The mathematical formulation of the real power injected at bus i:\n\n');
fprintf('   P_i^calc = sum_{j=1}^n ( V_i * V_j * Y_ij * cos(delta_i + delta_j - theta_ij) )\n\n');
fprintf('   V_i   = voltage magnitude at i\n');
fprintf('   V_j   = voltage magnitude at j\n');
fprintf('   Y_ij  = magnitude of admittance between bus i and j\n');
fprintf('   delta_i = voltage angle at bus i\n');
fprintf('   delta_j = voltage angle at bus j\n');
fprintf('   theta_ij = angle of admittance Y_ij\n\n');

fprintf('The mathematical formulation of the reactive power injected at bus i:\n\n');
fprintf('   Q_i^calc = sum_{j=1}^n ( V_i * V_j * Y_ij * sin(delta_i + delta_j - theta_ij) )\n\n');

fprintf('The mathematical formulation of the mismatch, requires us to take into consideration these factors, that is,:\n\n');
fprintf('   we''ll convert our specified power values (MW & MVAr) from the table into p.u.\n');

fprintf('   We use the S-base to convert all MW (P) and MVAr (Q) values:\n\n');
fprintf('   P_p.u = P_MW / S_base     and     Q_p.u = Q_MVAr / S_base\n');

fprintf('   P^spec = P_gen - P_load   and     Q^spec = Q_gen - Q_load\n\n');

fprintf('Then applying the above questions, we get the following:\n\n');

      for bus = 2:n_bus
         fprintf(' Bus %d: P_calc = %.6f pu, Q_calc = %.6f pu, ΔP = %.6e, ΔQ = %.6e\n', ...
            bus, P_calc(bus), Q_calc(bus), DeltaP(bus), DeltaQ(bus));
      end

%% ------------------------------------------------ LITERATURE RELATED TO Q LIMIT CHECKS ---------------------------------------
```

Figure 22: matlab code showing the declaration of system parameters (left) and calculation of mismatches for the 1st iteration

```matlab
    end

%% ---------------------------------------------- LITERATURE RELATED TO Q LIMIT CHECKS ----------------------------------------------

    fprintf('\n[STEP 8] REACTIVE POWER CHECKS\n\n');
    fprintf(['ASSUMED REACTIVE POWER LIMITS FOR BUS 2 GENERATOR\n\n', ...
            'Since the reactive power limits Qmin and Qmax were not explicitly provided for the generator\n', ...
            'at Bus 2, we adopt an industry-standard approximation commonly used in power flow studies.\n', ...
            'According to the Iowa State University EE553 Power Flow module, generator capability curves\n', ...
            'indicate that the available reactive power decreases as the real power output increases.\n', ...
            'In the absence of detailed capability curves, the document suggests modeling reactive limits\n', ...
            'using a conservative fixed band. Specifically, the limits are approximated by the formula:\n', ...
            'Qmin = -0.5 × P and Qmax = +0.5 × P, where P is the rated real power output.\n', ...
            'This simplification results in a rectangular reactive capability region around P.\n', ...
            '(see Fig. T7.1, "Generator Capability Curve and Approximate Reactive Limits").\n\n', ...
            'Reference:\n', ...
            'Iowa State University, EE 553: Power Flow Module. Available at:\n', ...
            'https://home.engineering.iastate.edu/~jdm/ee553/PowerFlow.pdf\n\n']);

    fprintf('Therefore conducting the Q-limit check on Bus 2:\n');
    fprintf(' Q2 = %.6f pu, Qmin = %.6f pu, Qmax = %.6f pu\n', Q_calc(2), Qmin_pu, Qmax_pu);
    if Q_calc(2) >= Qmin_pu && Q_calc(2) <= Qmax_pu
        fprintf(' Therefore, Q2 within the limits and as of such, the PV bus valid)\n');
    else
        fprintf('->Q2 outside limits (treat as PQ or adjust)\n');
    end

    fprintf('\nCurrent max mismatch = %.6e\n', maxMismatch);
  else
    % Compact progress line
    fprintf('Iter %d: max mismatch = %.6e\n', iter, maxMismatch);
  end

%% ---------------------------------------------END LITERATURE RELATED TO Q LIMIT CHECKS ---------------------------------------------
```

```matlab
%% -----------------------------------------------------COMPUTING THE JACOBIAN MATRIX -----------------------------------------------------

    %% --- Step B: Convergence check BEFORE Jacobian
    if maxMismatch < eps_tol
        converged = true;
        if iter == 1 && show_full_detail_first_iter
            fprintf('\n[CONVERGED BEFORE UPDATE] Iteration %d: max mismatch = %.6e < eps_tol\n', iter, maxMismatch);
        end
        break;
    end

    %% --- Step C: Compute Jacobian J
    delta = angle(V); Vm = abs(V);
    G = real(Ybus); B = imag(Ybus);
    PQ    = [3 4];
    PV_PQ = [2 3 4];
    nPQ = length(PQ); nAngle = length(PV_PQ);
    J1 = zeros(nAngle); J2 = zeros(nAngle, nPQ);
    J3 = zeros(nPQ, nAngle); J4 = zeros(nPQ);

    for i = 1:nAngle
        m = PV_PQ(i);
        for j = 1:nAngle
            n = PV_PQ(j);
            if m == n
                sum_term = 0;
                for k = 1:n_bus
                    if k ~= m
                        sum_term = sum_term + Vm(m)*Vm(k)*(G(m,k)*sin(delta(m)-delta(k)) - B(m,k)*cos(delta(m)-delta(k)));
                    end
                end
                J1(i,j) = -sum_term;
            else
                J1(i,j) = Vm(m)*Vm(n)*(G(m,n)*sin(delta(m)-delta(n)) - B(m,n)*cos(delta(m)-delta(n)));
            end
        end
    end

    for i = 1:nAngle
        m = PV_PQ(i);
        for j = 1:nPQ
            n = PQ(j);
            if m == n
                sum_term = 0;
                for k = 1:n_bus
                    sum_term = sum_term + Vm(k)*(G(m,k)*cos(delta(m)-delta(k)) + B(m,k)*sin(delta(m)-delta(k)));
                end
                J2(i,j) = 2*Vm(m)*sum_term;
            else
                J2(i,j) = Vm(m)*(G(m,n)*cos(delta(m)-delta(n)) + B(m,n)*sin(delta(m)-delta(n)));
            end
        end
    end

    for i = 1:nPQ
        m = PQ(i);
        for j = 1:nAngle
            n = PV_PQ(j);
            if m == n
                sum_term = 0;
                for k = 1:n_bus
                    if k ~= m
                        sum_term = sum_term + Vm(m)*Vm(k)*(G(m,k)*cos(delta(m)-delta(k)) + B(m,k)*sin(delta(m)-delta(k)));
                    end
```

Figure 23: matlab code showing the reactive power limit checks (left) and the computation of the Jacobian Matrix (right) for the 1st iteration

```
            sum_term = 0;
            for k = 1:n_bus
                sum_term = sum_term + Vm(k)*(G(m,k)*cos(delta(m)-delta(k)) + B(m,k)*sin(delta(m)-delta(k)));
            end
            J2(i,j) = 2*Vm(m)*sum_term;
        else
            J2(i,j) = Vm(m)*(G(m,n)*cos(delta(m)-delta(n)) + B(m,n)*sin(delta(m)-delta(n)));
        end
    end
end

for i = 1:nPQ
    m = PQ(i);
    for j = 1:nAngle
        n = PV_PQ(j);
        if m == n
            sum_term = 0;
            for k = 1:n_bus
                if k ~= m
                    sum_term = sum_term + Vm(m)*Vm(k)*(G(m,k)*cos(delta(m)-delta(k)) + B(m,k)*sin(delta(m)-delta(k)));
                end
            end
            J3(i,j) = sum_term;
        else
            J3(i,j) = -Vm(m)*Vm(n)*(G(m,n)*cos(delta(m)-delta(n)) + B(m,n)*sin(delta(m)-delta(n)));
        end
    end
end

for i = 1:nPQ
    m = PQ(i);
    for j = 1:nPQ
        n = PQ(j);
        if m == n
            sum_term = 0;
            for k = 1:n_bus
                sum_term = sum_term + Vm(k)*(G(m,k)*sin(delta(m)-delta(k)) - B(m,k)*cos(delta(m)-delta(k)));
            end
            J4(i,j) = -2*Vm(m)*B(m,m) - sum_term;
        else
            J4(i,j) = -Vm(m)*(G(m,n)*sin(delta(m)-delta(n)) - B(m,n)*cos(delta(m)-delta(n)));
        end
    end
end
J = [J1 J2; J3 J4];

% ---- Jacobian literature EXACTLY as given ----
if iter == 1 && show_full_detail_first_iter
   fprintf('\n---BEFORE GOING TO STEP C, HERE ARE NOTES RELATED TO THE JACOBIAN---\n');
    fprintf('\nJacobian Matrix [ΔP; ΔQ] = J × [Δδ; ΔV]:\n\n');
    fprintf('State vector: x = [ δ2 δ3 δ4 |V3| |V4| ]\n');
    fprintf('Mismatch vec: Δ = [ ΔP2 ΔP3 ΔP4 ΔQ3 ΔQ4 ]\n\n');
    fprintf('Jacobian layout: \n\n');
    fprintf('[ ∂ΔP2/∂δ2 ∂ΔP2/∂δ3 ∂ΔP2/∂δ4 ∂ΔP2/∂|V3| ∂ΔP2/∂|V4| ]\n');
    fprintf('[ ∂ΔP3/∂δ2 ∂ΔP3/∂δ3 ∂ΔP3/∂δ4 ∂ΔP3/∂|V3| ∂ΔP3/∂|V4| ]\n');
    fprintf('[ ∂ΔP4/∂δ2 ∂ΔP4/∂δ3 ∂ΔP4/∂δ4 ∂ΔP4/∂|V3| ∂ΔP4/∂|V4| ]\n');
    fprintf('[ ∂ΔQ3/∂δ2 ∂ΔQ3/∂δ3 ∂ΔQ3/∂δ4 ∂ΔQ3/∂|V3| ∂ΔQ3/∂|V4| ]\n');
    fprintf('[ ∂ΔQ4/∂δ2 ∂ΔQ4/∂δ3 ∂ΔQ4/∂δ4 ∂ΔQ4/∂|V3| ∂ΔQ4/∂|V4| ]\n\n');

    fprintf('[Step C] NUMERICAL JACOBIAN (J):\n\n');
    disp(J);
end

%% ------------------------------------------------------ END COMPUTING THE JACOBIAN MATRIX ---------------------------------
```

```
%% --------------------------------------------------------- COMPUTING THE MISMATCHES ---------------


    %% --- Step D: Solve J * dx = mism (Cramer's rule)
    mism = [DeltaP(2); DeltaP(3); DeltaP(4); DeltaQ(3); DeltaQ(4)];
    [nJrow, nJcol] = size(J);
    if nJrow ~= nJcol, error('Jacobian J must be square.'); end
    if length(mism) ~= nJrow, error('Mismatch vector length must equal Jacobian size.'); end

    detJ = det(J);
    if iter == 1 && show_full_detail_first_iter
        fprintf('\n[Step D] SOLVING J Δx = Δ USING Cramer''s RULE (dx values not yet applied):\n\n');
        fprintf(' det(J) = %.6e\n', detJ);
        if abs(detJ) < 1e-12
            warning(' det(J) is extremely small — Cramer''s rule may be numerically unstable.');
        end
    end

    dx = zeros(nJrow,1);
    for col = 1:nJcol
        Ji = J; Ji(:,col) = mism;
        detJi = det(Ji);
        dx(col) = detJi / detJ;
        if iter == 1 && show_full_detail_first_iter
            fprintf(' dx(%d) = %.6e\n', col, dx(col));
        end
    end

    % Map dx into state increments
    d_delta2 = dx(1); d_delta3 = dx(2); d_delta4 = dx(3);
    d_V3     = dx(4); d_V4     = dx(5);

    if iter == 1 && show_full_detail_first_iter
        fprintf('\n[Step D] SOLUTION FOR THE STATE VECTORS:\n\n');
        fprintf(' dδ2 = %.8f rad, dδ3 = %.8f rad, dδ4 = %.8f rad\n', d_delta2, d_delta3, d_delta4);
        fprintf(' d|V3| = %.8f pu, d|V4| = %.8f pu\n', d_V3, d_V4);
    end

%% --------------------------------------------------------END COMPUTING THE MISMATCHES ------------



%% --------------------------------------------------------WORKING AROUND THE MISMATCHES -----------

    %% --- Step E: Update state variables
    delta = angle(V); Vm = abs(V);
    delta(2) = delta(2) + d_delta2;
    delta(3) = delta(3) + d_delta3;
    delta(4) = delta(4) + d_delta4;
    Vm(3)    = Vm(3)  + d_V3;
    Vm(4)    = Vm(4)  + d_V4;

    for b = 1:n_bus
        V(b) = Vm(b) * exp(1j*delta(b));
    end
```

Figure 24: continuation of the computation of the Jacobian Matrix (left) and mismatches (right) for the 1st iteration

```
%% -----------------------------------------------------------WORKING AROUND THE MISMATCHES ---------

    %% --- Step E: Update state variables
    delta = angle(V); Vm = abs(V);
    delta(2) = delta(2) + d_delta2;
    delta(3) = delta(3) + d_delta3;
    delta(4) = delta(4) + d_delta4;
    Vm(3)    = Vm(3)   + d_V3;
    Vm(4)    = Vm(4)   + d_V4;

    for b = 1:n_bus
        V(b) = Vm(b) * exp(1j*delta(b));
    end

    if iter == 1 && show_full_detail_first_iter
        fprintf('\n[Step E] UPDATED STATE AFTER APPLYING dx:\n\n');
        for b = 1:n_bus
            fprintf(' Bus %d: |V| = %.6f pu, angle = %.6f rad (%.6f°)\n', b, abs(V(b)), angle(V(b)),
        end
    end

    %% --- Step F: Recompute mismatches (verification) after update
    DeltaP_new = zeros(n_bus,1); DeltaQ_new = zeros(n_bus,1);
    P_calc_new = zeros(n_bus,1); Q_calc_new = zeros(n_bus,1);

    if iter == 1 && show_full_detail_first_iter
        fprintf('\n[Step F] VERIFICATION: RECOMPUTING INJECTIONS WITH UPDATING V\n\n');
        fprintf('We must also alter our previous approximations for δ2, δ3, |V2|\n');
        fprintf('Unknown Value_{new} = Unknown Value_{old} + Solved Δvalue\n\n');

    end

    for bus = 2:n_bus
        I = Ybus(bus,:) * V.';
        S = V(bus) * conj(I);
        P_calc_new(bus) = real(S);
        Q_calc_new(bus) = imag(S);
        DeltaP_new(bus) = P_spec(bus) - P_calc_new(bus);
        DeltaQ_new(bus) = Q_spec(bus) - Q_calc_new(bus);

        if iter == 1 && show_full_detail_first_iter
            fprintf(' Bus %d: P_new = %.6f, Q_new = %.6f, ΔP_new = %.6e, ΔQ_new = %.6e\n', ...
                bus, P_calc_new(bus), Q_calc_new(bus), DeltaP_new(bus), DeltaQ_new(bus));
        end
    end

    mism_vec_new    = [DeltaP_new(2); DeltaP_new(3); DeltaP_new(4); DeltaQ_new(3); DeltaQ_new(4)];
    maxMismatch_new = max(abs(mism_vec_new));

%% -----------------------------------------------------------END WORKING AROUND THE MISMATCHES -----
```

```
%% ------------------------------------------------------------CONVERGENCE LITERATURE ------------------------------------------------------------

    % ---- Convergence literature ----
    if iter == 1 && show_full_detail_first_iter
        fprintf('\nNOW WE NEED TO CONDUCT THE CONVERGENCE CHECK  \n');

        fprintf(['\nSince the Newton-Raphson method relies on multiple iterations to solve the\n' ...
                 'nonlinear power flow equations, the solutions are just approximations and\n' ...
                 'there needs to be a convergence check in order to determine if the solution\n' ...
                 'is within the acceptable tolerances.\n\n']);

        fprintf(['To do this, we need to formulate a well-defined stopping criterion, that is,\n' ...
                 'a well-defined epsilon (ε) value:\n\n']);

        fprintf(['According to the IEEE Standard 3002.2-2018 ("Recommended Practice for Conducting\n' ...
                 'Load-Flow Studies and Analysis of Industrial and Commercial Power Systems"), the\n' ...
                 'power mismatch tolerance is generally specified in the range of 0.01 to 0.0001\n' ...
                 'per unit on the system MVA base.\n' ...
                 'Reference: https://www.researchgate.net/profile/Debrayan-Bravo-Hidalgo/post/\n' ...
                 'PV_system_power_generated/attachment/61ae8830b3729f0f61984617/AS%%3A1098138710478849%%401638828079845/\n' ...
                 'download/IEEE+Recommended+Practice+for+Conducting+Load-Flow+Studies+and+Analysis+of+Industrial+and+\n' ...
                 'Commercial+Power+Systems.pdf\n\n']);

        fprintf('Therefore, in our case, the iterative process will be considered converged when ε is 1e-6:\n\n');
        fprintf('max_i (|%cP_i|, |%cQ_i|) < %.0e\n\n', char(916), char(916), 1e-6);
    end

    % Post-Update Convergence Check (after Step F)
    if maxMismatch_new < eps_tol
        converged = true;
        fprintf('Converged at iter %d: max mismatch = %.6e < eps_tol\n', iter, maxMismatch_new);
        break;
    end
    % loop continues with compact line already printed at top
end % for iter

%% ------------------------------------------------------------END CONVERGENCE LITERATURE ------------------------------------------------------------
```

Figure 25: continuation of computing the mismatches (top) for the 1st iteration and convergence test related literature (bottom)

```
%% ---------------------------------------------------------------------------------------------------------------

%%                                                      FINAL RESULTS & SUMMARY
%% ---------------------------------------------------------------------------------------------------------------
if ~converged
    fprintf('\nDid not converge within %d iterations. Final max mismatch = %.6e\n', max_iter, maxMismatch_new);
    warning('Did not converge within %d iterations. Final max mismatch = %.6e', max_iter, maxMismatch_new);
else
    fprintf('\nConverged in %d iterations. Final max mismatch = %.6e\n', iter, maxMismatch_new);
end

fprintf('\n[FINAL BUS VOLTAGES]\n');
for b = 1:n_bus
    fprintf(' Bus %d: |V| = %.6f pu, angle = %.6f rad (%.6f°)\n', b, abs(V(b)), angle(V(b)), angle(V(b))*180/pi);
end

% (Iteration history summary intentionally omitted)

% --- Bus power injections from final V ---
S = zeros(1, n_bus);
for k = 1:n_bus
    S(k) = V(k) * conj(Ybus(k,:) * V.');  % Vk * conj(sum_j Y_kj * Vj)
end
P = real(S);
Q = imag(S);

fprintf('\n[BUS INJECTIONS]\n');
for k = 1:n_bus
    fprintf('Bus %d: P = %.6f pu (%.3f MW), Q = %.6f pu (%.3f MVAr)\n', ...
        k, P(k), P(k)*S_base, Q(k), Q(k)*S_base);
end  % <-- THIS was missing

% (optional quick check)
fprintf('\n[POWER BALANCE CHECK] ΣP = %.6f pu (%.3f MW), ΣQ = %.6f pu (%.3f MVAr)\n', ...
        sum(P), sum(P)*S_base, sum(Q), sum(Q)*S_base);


%% ---------------------------------- PART B: SHORT CIRCUIT STUDY (IEC 60909) -------------------------------
```
Figure 26: matlab code snippet showing the final bus voltages and power injected for the 1st iteration

```matlab
fprintf('\n\n=====================================================================
fprintf('                        PART B: SHORT CIRCUIT STUDY (IEC 60909)\n');
fprintf('=====================================================================

% ------------------------ User Inputs (EDIT to your study) ------------------------
target_bus     = 1;          % We only report results for THIS bus (BUS 1)
Un_kV          = 132;        % nominal L-L voltage [kV] (scalar or 1×n_bus)
c_max          = 1.10;       % IEC voltage factor for MAX Ik"
c_min          = 1.00;       % IEC voltage factor for MIN Ik"
Zf_ohm         = complex(20,40);  % GIVEN fault impedance: 20 + j40 ohm (series at fault)
R_scale_min    = 1.20;       % increase only R for MIN case (hot conductors)
Ssc_source_MVA  = 7200;      % grid short-circuit power behind slack (Bus 1)

% ------------------------ Voltage vector in volts ------------------------
if isscalar(Un_kV)
   Un_V = repmat(Un_kV*1e3, 1, n_bus);
else
   Un_V = Un_kV(:).'*1e3;
   if numel(Un_V) ~= n_bus
      error('Un_kV must be scalar or length n_bus (%d).', n_bus);
   end
end

% ------------------------ Sanity: finite source level --------------------
if ~(isnumeric(Ssc_source_MVA) && isfinite(Ssc_source_MVA) && Ssc_source_MVA > 0)
   warning('Ssc_source_MVA must be > 0. Using 3000 MVA as default.');
   Ssc_source_MVA = 3000;
end

% ------------------------ Build SC Y-bus: MAX case -----------------------
Ybus_sc_max = zeros(n_bus);
for kk = 1:size(line_data,1)
   i = line_data(kk,1); j = line_data(kk,2);
   R = line_data(kk,3); X = line_data(kk,4);
   Zij = complex(R,X); yij = 1/Zij;
   Ybus_sc_max(i,j) = Ybus_sc_max(i,j) - yij;
   Ybus_sc_max(j,i) = Ybus_sc_max(j,i) - yij;
   Ybus_sc_max(i,i) = Ybus_sc_max(i,i) + yij;
   Ybus_sc_max(j,j) = Ybus_sc_max(j,j) + yij;
end

% ------------------------ Build SC Y-bus: MIN case (R hotter) ------------
Ybus_sc_min = zeros(n_bus);
for kk = 1:size(line_data,1)
   i = line_data(kk,1); j = line_data(kk,2);
   R = line_data(kk,3)*R_scale_min; X = line_data(kk,4);
   Zij = complex(R,X); yij = 1/Zij;
   Ybus_sc_min(i,j) = Ybus_sc_min(i,j) - yij;
   Ybus_sc_min(j,i) = Ybus_sc_min(j,i) - yij;
   Ybus_sc_min(i,i) = Ybus_sc_min(i,i) + yij;
   Ybus_sc_min(j,j) = Ybus_sc_min(j,j) + yij;
end

% ------------------------ Add Thevenin source at Slack (Bus 1) -----------
% Zs(ohm) = (kV^2)/MVA   [uses Bus-1 nominal voltage]
Zs_ohm = ( (Un_V(1)/1e3)^2 ) / Ssc_source_MVA;
Ybus_sc_max(1,1) = Ybus_sc_max(1,1) + 1/Zs_ohm;
Ybus_sc_min(1,1) = Ybus_sc_min(1,1) + 1/Zs_ohm;

% ------------------------ Thevenin impedances (Z-bus diagonals) ----------
Zbus_max = inv(Ybus_sc_max);
Zbus_min = inv(Ybus_sc_min);
```

Figure 27: matlab code snippet showing the short circuit procedure on bus 1

```
  R = line_data(kk,3)*R_scale_min; X = line_data(kk,4);
  Zij = complex(R,X); yij = 1/Zij;
  Ybus_sc_min(i,j) = Ybus_sc_min(i,j) - yij;
  Ybus_sc_min(j,i) = Ybus_sc_min(j,i) - yij;
  Ybus_sc_min(i,i) = Ybus_sc_min(i,i) + yij;
  Ybus_sc_min(j,j) = Ybus_sc_min(j,j) + yij;
end

% -------------------------- Add Thevenin source at Slack (Bus 1) ------------
% Zs(ohm) = (kV^2)/MVA   [uses Bus-1 nominal voltage]
Zs_ohm = ( (Un_V(1)/1e3)^2 ) / Ssc_source_MVA;
Ybus_sc_max(1,1) = Ybus_sc_max(1,1) + 1/Zs_ohm;
Ybus_sc_min(1,1) = Ybus_sc_min(1,1) + 1/Zs_ohm;

% -------------------------- Thevenin impedances (Z-bus diagonals) -----------
Zbus_max = inv(Ybus_sc_max);
Zbus_min = inv(Ybus_sc_min);
Zth_max  = diag(Zbus_max).';
Zth_min  = diag(Zbus_min).';

% -------------------------- Bus-1 totals incl. fault impedance -------------
Ztot_max_bus1 = Zth_max(target_bus) + Zf_ohm;
Ztot_min_bus1 = Zth_min(target_bus) + Zf_ohm;

% -------------------------- IEC 60909 initial Ik" at Bus 1 -----------------
Ik_max_bus1_A = (c_max * Un_V(target_bus)) / (sqrt(3) * abs(Ztot_max_bus1));
Ik_min_bus1_A = (c_min * Un_V(target_bus)) / (sqrt(3) * abs(Ztot_min_bus1));

% Short-circuit powers
Sk_max_bus1_MVA = (sqrt(3) * Un_V(target_bus) * Ik_max_bus1_A) / 1e6;
Sk_min_bus1_MVA = (sqrt(3) * Un_V(target_bus) * Ik_min_bus1_A) / 1e6;

% -------------------------------- REPORT ----------------------------------
fprintf('\nINPUTS\n');
fprintf('  Target bus: %d\n', target_bus);
fprintf('  Un (kV): %.3f  |  c_max: %.2f   c_min: %.2f\n', Un_V(target_bus)/1e3, c_max, c_min);
fprintf('  Zf (ohm): %.3f%+.3fj  |  Ssc at slack (MVA): %.1f  |  R_scale_min: %.2f\n', ...
     real(Zf_ohm), imag(Zf_ohm), Ssc_source_MVA, R_scale_min);
fprintf('  Zs at slack (ohm): %.6f\n', Zs_ohm);

fprintf('\nTHEVENIN EQUIVALENTS (at Bus %d)\n', target_bus);
fprintf('  Z_th MAX: % .6f %+.6fj  ohm\n', real(Zth_max(target_bus)), imag(Zth_max(target_bus)));
fprintf('  Z_th MIN: % .6f %+.6fj  ohm\n', real(Zth_min(target_bus)), imag(Zth_min(target_bus)));
fprintf('  Z_total incl. Zf (MAX): % .6f %+.6fj  ohm\n', real(Ztot_max_bus1), imag(Ztot_max_bus1));
fprintf('  Z_total incl. Zf (MIN): % .6f %+.6fj  ohm\n', real(Ztot_min_bus1), imag(Ztot_min_bus1));

fprintf('\nRESULTS (BUS %d ONLY)\n', target_bus);
fprintf('+-------------------------+-----------+-------------+-------------+\n');
fprintf('| Quantity                |  Unit    |  MAX case  |  MIN case  |\n');
fprintf('+-------------------------+-----------+-------------+-------------+\n');
fprintf('| Ik" (initial 3phi)      |   kA     | %12.3f | %12.3f |\n', Ik_max_bus1_A/1e3, Ik_min_bus1_A/1e3);
fprintf('| Sk" (short-circ power)  |   MVA    | %12.3f | %12.3f |\n', Sk_max_bus1_MVA,  Sk_min_bus1_MVA);
fprintf('+-------------------------+-----------+-------------+-------------+\n');

% R/X (useful if later computing peak i_p = kappa*sqrt(2)*Ik")
Rx_max = real(Zth_max(target_bus)); Xx_max = imag(Zth_max(target_bus));
fprintf('\nR/X at Bus %d (MAX-case Z_th): %.4f\n', target_bus, Rx_max / max(eps, abs(Xx_max)));
fprintf('--------------------------------------------------------------------------------\n');
```

Figure 28: continuation: matlab code snippet showing the short circuit procedure on bus 1

Appendix B: MATLAB Simulink Model Screenshots and Configuration Settings

It is important to classify the buses as either slack, PV, or PQ buses before they can be configured. In the given system, bus 1 is a slack bus, bus 2 is a PV bus and buses 3 and 4 are PQ buses. Different components such as generators, transformers, transmission lines, and loads are represented by preset blocks found in the library.

Physical System Component	*Simulink Block Model*
Generator	Three-Phase Source Block
Buses	Three-Phase V-I Measurement
Transmission Line	Three-Phase Series RLC Branch
Load	Three-Phase Parallel RLC Load

Load Flow Bus Block helps with the calculations of the load flow analysis. Actual values are used when configuring transmission line. If the transmission line in the given system is in per unit, it must be converted to actual values using base values.

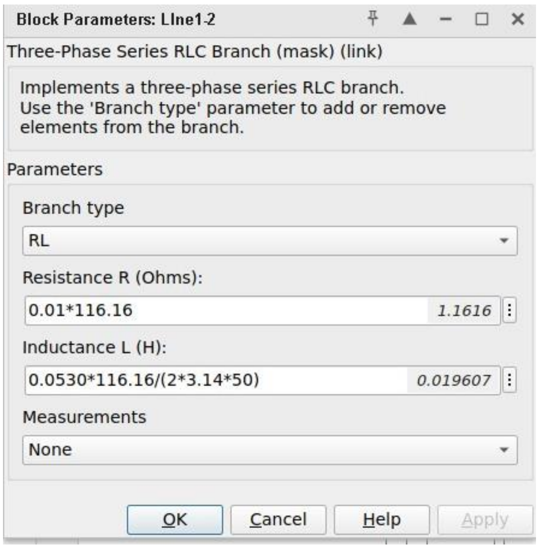


Figure 29: Transmission line configuration

Powergui block helps analyse and simulate power systems. It must be set to phasor mode, this helps perform the load flow analysis accurately.

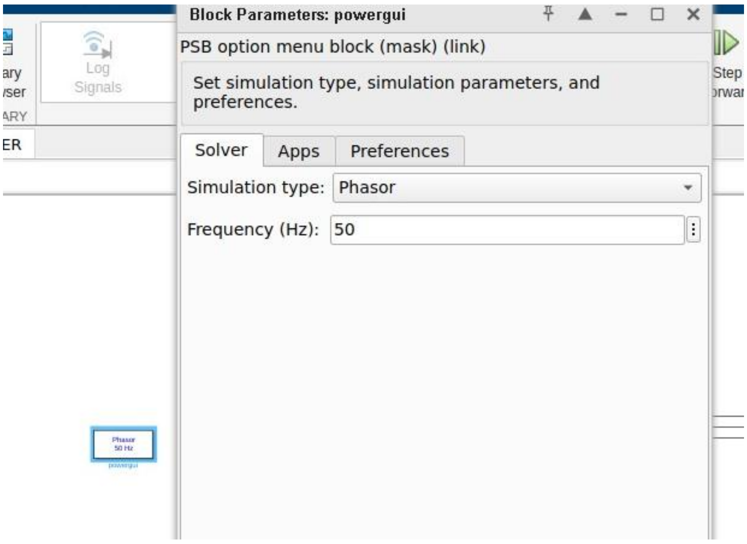


Figure 30: powergui block configuration.

The powergui block also allows verification of preset values, before computation. This ensures that the simulation settings match with the system parameters.

Model: POWER | Update | The table shows the load flow settings of the model. Click Compute to perform load flow analysis. | Compute | Apply | Add bus blocks | Report

Type: Positive sequence load flow

	Block name	Block type	Bus type	Bus ID	Vbase (kV)	Vref (pu)	Vangle (deg)	P (MW)	Q (Mvar)	Qmin (Mvar)	Qmax (Mvar)	V_LF (pu)	Vangle_LF (deg)	P_LF (MW)	Q_LF (Mvar)
1	SLack	Vsrc	swing	BUS_1	132.0000	1.0100	0	0.0100	0	-Inf	Inf	0	0	0	0
2	Load2	RLC load	PQ	BUS_2	132.0000	1.0000	0	75.0000	50.0000	-Inf	Inf	0	0	0	0
3	g2	Vsrc	PV	BUS_2	132.0000	1.0000	0	100.0000	0	-Inf	Inf	0	0	0	0
4	Load3	RLC load	PQ	BUS_3	132.0000	1.0000	0	79.0000	25.0000	-Inf	Inf	0	0	0	0
5	Load4	RLC load	PQ	BUS_4	132.0000	1.0000	0	85.0000	40.0000	-Inf	Inf	0	0	0	0

Figure 31: The preset values on Simulink before being computed.

The following table provides a summary of the simulation's convergence:

	Block name	Block type	Bus type	Bus ID	Vbase (kV)	Vref (pu)	Vangle (deg)	P (MW)	Q (Mvar)	Qmin (Mvar)	Qmax (Mvar)	V_LF (pu)	Vangle_LF (deg)	P_LF (MW)	Q_LF (Mvar)
1	SLack	Vsrc	swing	BUS_1	132.0000	1.0100	0	0.0100	0	-Inf	Inf	1.0100	0	139.9200	88.5104
2	Load2	RLC load	PQ	BUS_2	132.0000	1.0000	0	75.0000	50.0000	-Inf	Inf	1.0000	-0.1845	75.0000	50.0000
3	g2	Vsrc	PV	BUS_2	132.0000	1.0000	0	100.0000	0	-Inf	Inf	1.0000	-0.1845	100.0000	30.5422
4	Load3	RLC load	PQ	BUS_3	132.0000	1.0000	0	79.0000	25.0000	-Inf	Inf	0.9955	-1.1423	79.0000	25.0000
5	Load4	RLC load	PQ	BUS_4	132.0000	1.0000	0	85.0000	40.0000	-Inf	Inf	0.9951	-0.9259	85.0000	40.0000

Figure 32: Simulink results.

Patterns and comparisons:

- System stability was confirmed by the voltage magnitudes across all busses staying within allowable bounds (0.9985 – 1.01 pu).
- Losses and load demand throughout the network were offset by reactive power generation at the swing bus (88.51 Mvar).
- The outcomes validate the accuracy of the Newton-Raphson implementation and the Simulink model by closely matching the expected behaviour of the IEEE 4-bus system under balanced conditions.

Appendix C: DIgSILENT PowerFactory Model Screenshots and Configuration Settings

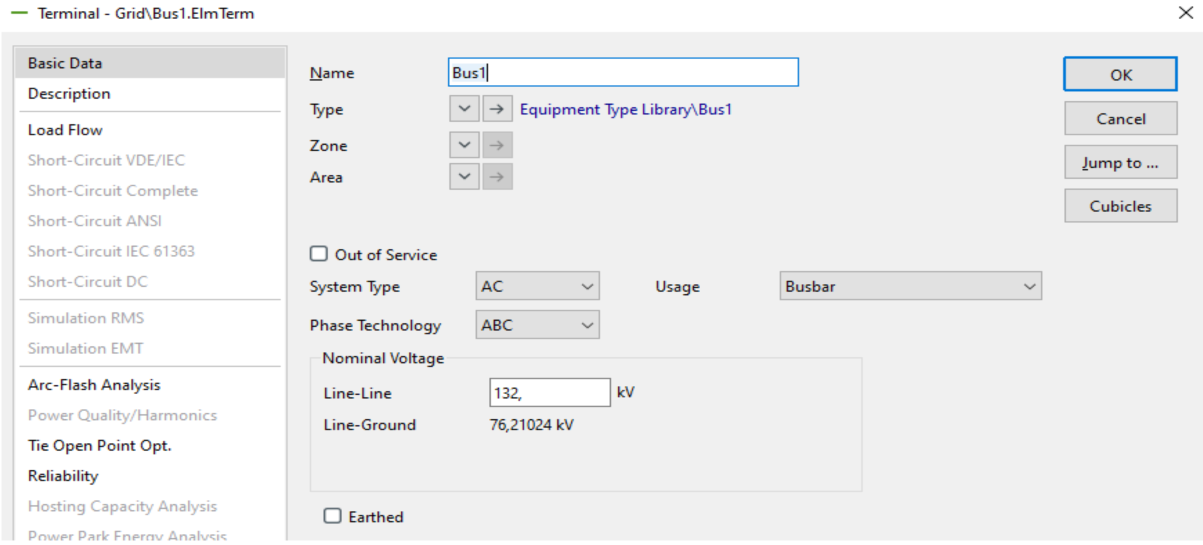
Section A: Configuring Bus'es, Loads, and Grids

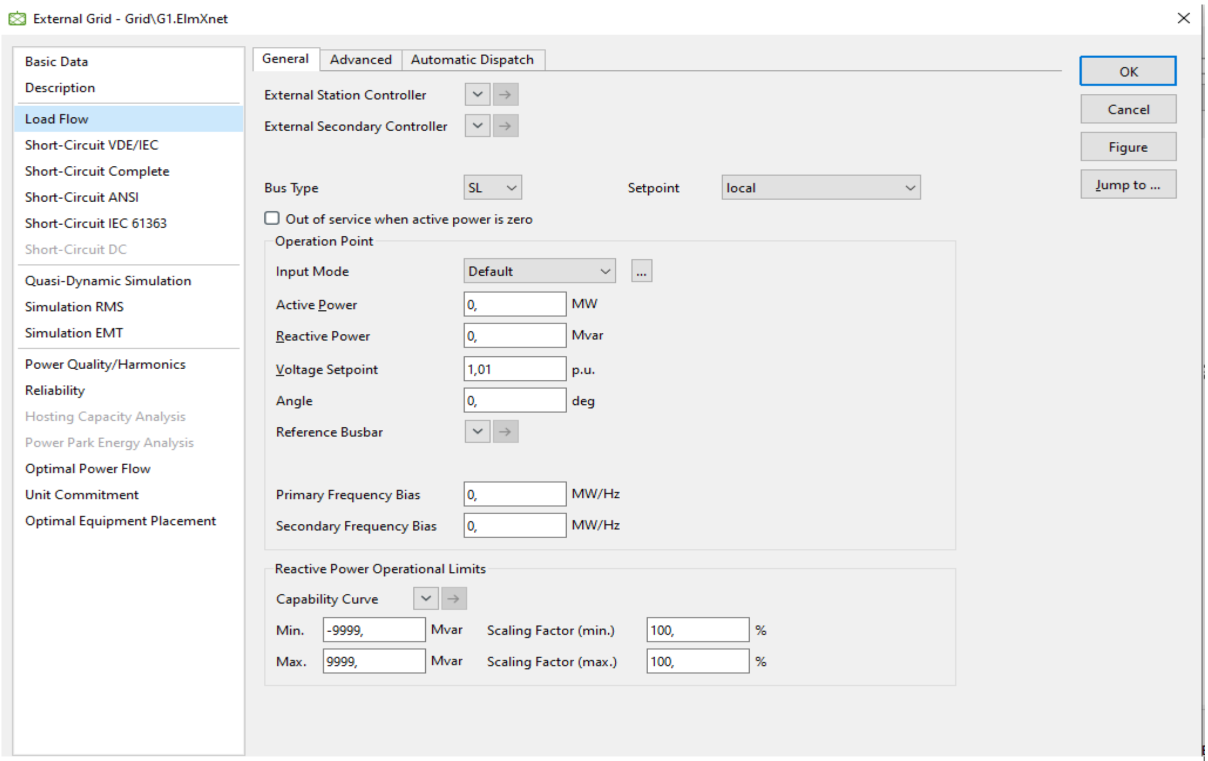
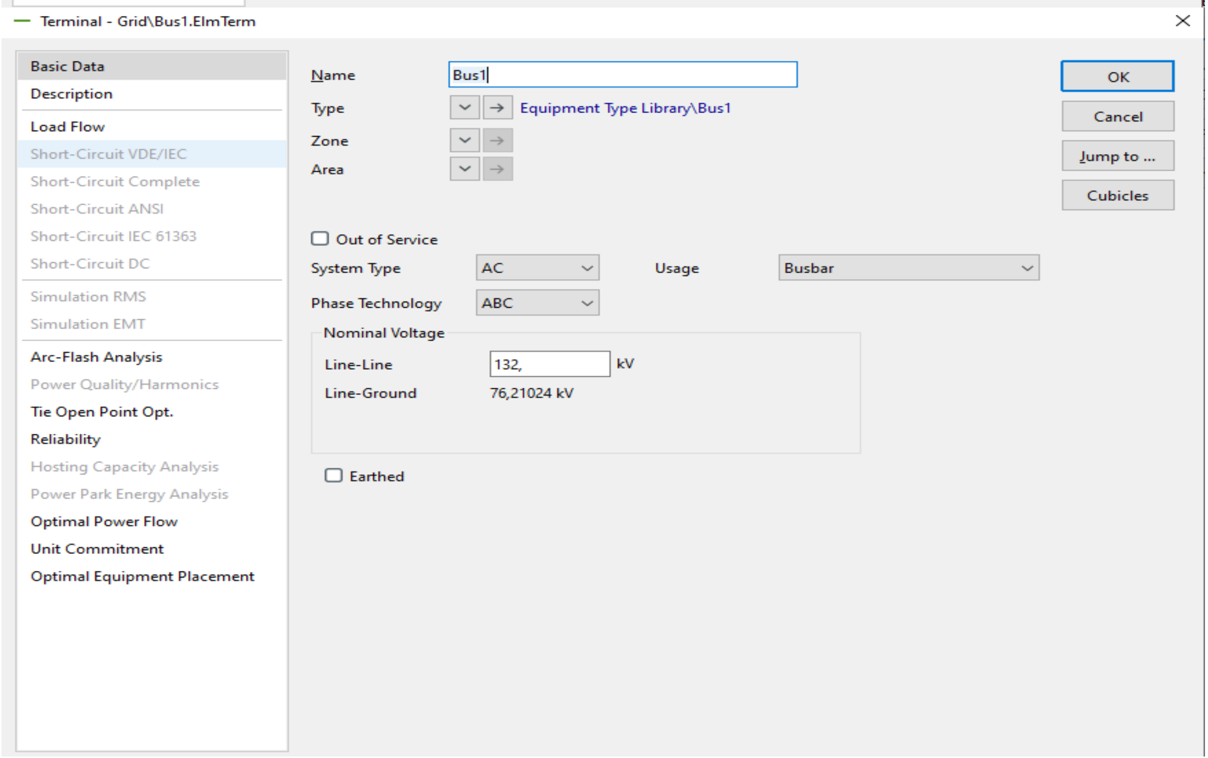
This appendix provides screenshots from DIgSILENT PowerFactory showing how the IEEE 4-bus test system was implemented and configured. The configuration process involved:

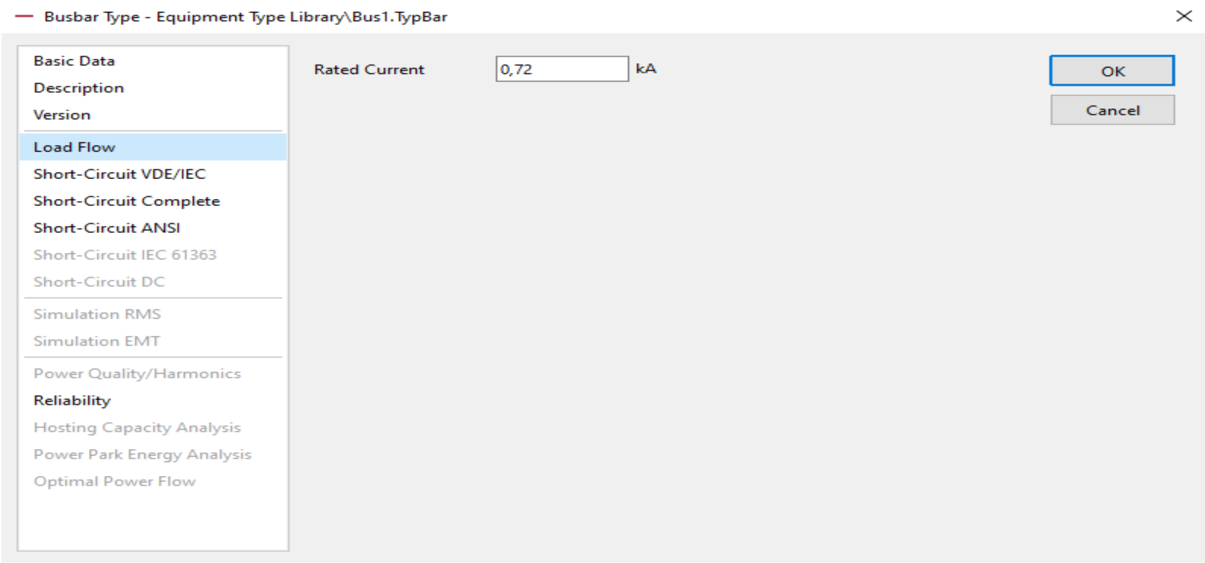
- Bus setup: Defining four buses at 132 kV nominal voltage, with one designated as the slack bus, one as a PV bus, and the remaining as PQ buses in line with the project brief.
- Grid and sources: Adding the external grid equivalent at the slack bus to represent the reference and specifying generator settings at the PV bus.
- Loads: Assigning active and reactive power demands at PQ buses according to the IEEE 4-bus data.
- Lines and transformers: Entering line impedances and transformer data based on the given system parameters, converted into per-unit on a 150 MVA, 132 kV base.

The following figures illustrate the system diagram, element properties, and calculation settings used in DIgSILENT PowerFactory for both load flow and short-circuit studies.

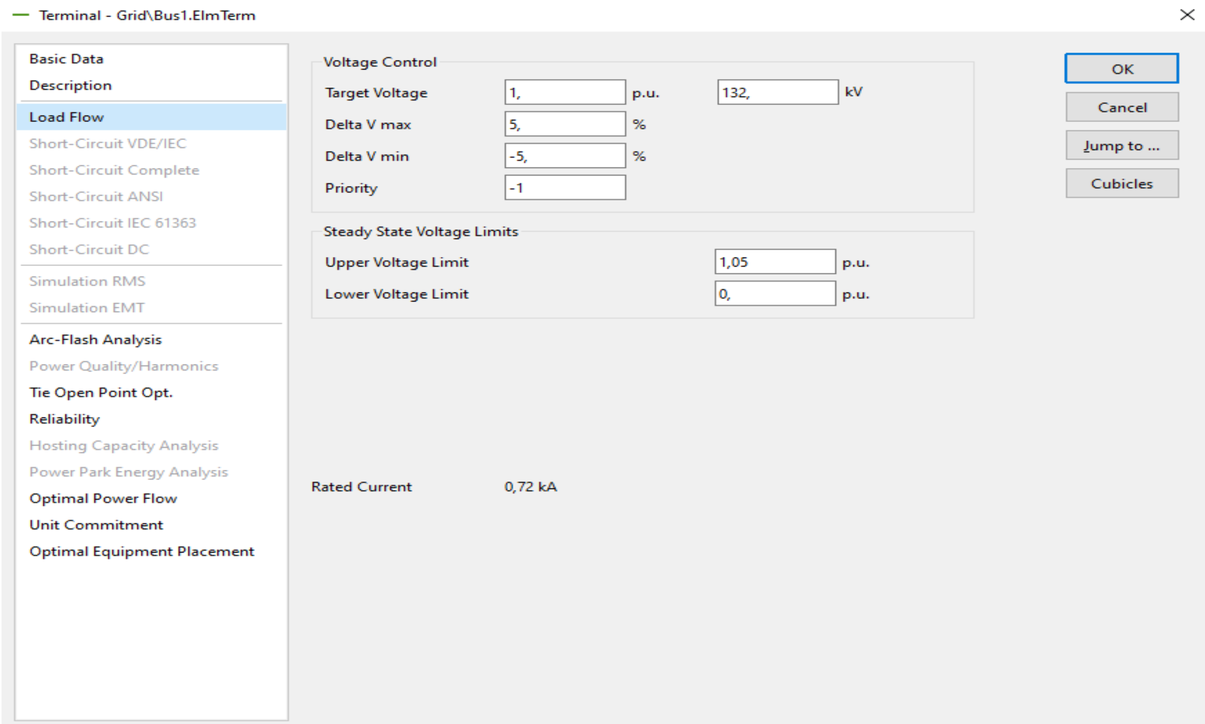
1. **Configuring the grid (G1)**

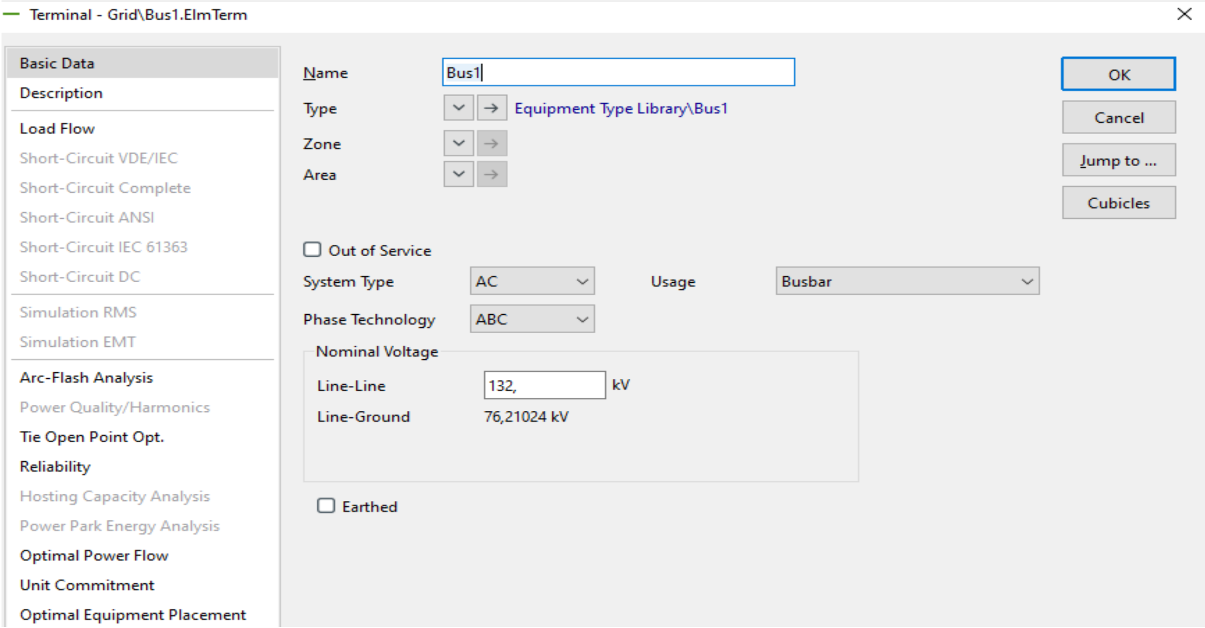


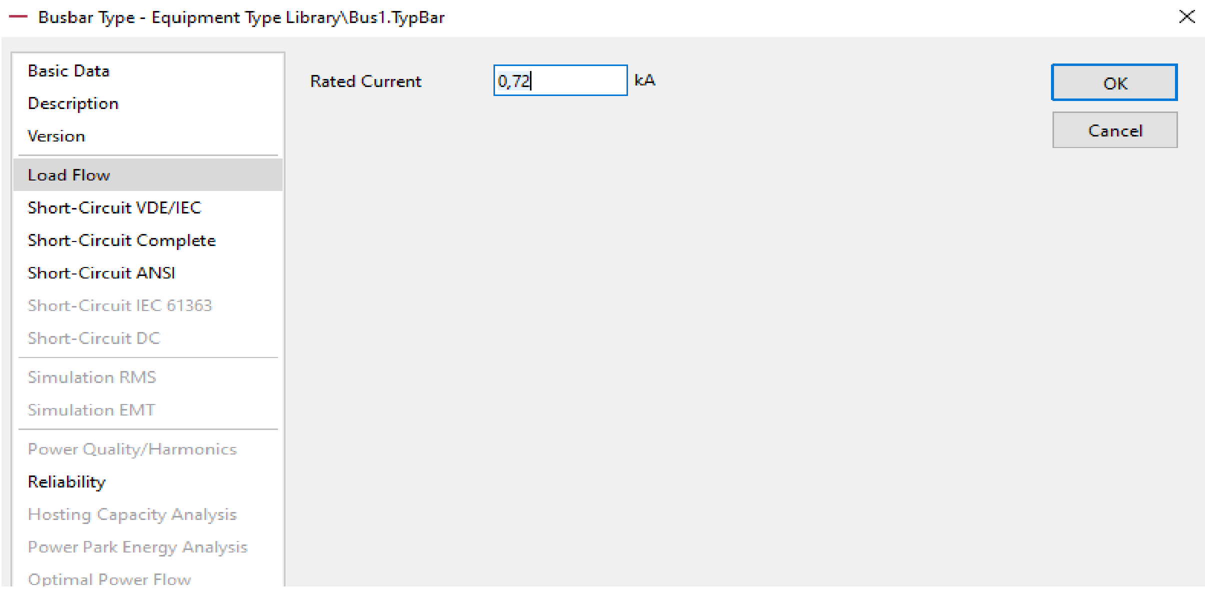

External Grid - Grid\G1.ElmXnet
Basic Data
Description
Load Flow
Short-Circuit VDE/IEC
Short-Circuit Complete
Short-Circuit ANSI
Short-Circuit IEC 61363
Short-Circuit DC
Quasi-Dynamic Simulation
Simulation RMS
Simulation EMT
Power Quality/Harmonics
Reliability
Hosting Capacity Analysis
Power Park Energy Analysis
Optimal Power Flow
Unit Commitment
Optimal Equipment Placement
General
Advanced
Automatic Dispatch
External Station Controller
External Secondary Controller
Bus Type
SL
Setpoint
local
Out of service when active power is zero
Operation Point
Input Mode
Default
Active Power
0,
MW
Reactive Power
0,
Mvar
Voltage Setpoint
1,01
p.u.
Angle
0,
deg
Reference Busbar
Primary Frequency Bias
0,
MW/Hz
Secondary Frequency Bias
0,
MW/Hz
Reactive Power Operational Limits
Capability Curve
Min.
-9999,
Mvar
Scaling Factor (min.)
100,
%
Max.
9999,
Mvar
Scaling Factor (max.)
100,
%
OK
Cancel
Figure
Jump to ...

Terminal - Grid\Bus1.ElmTerm
Basic Data
Description
Load Flow
Short-Circuit VDE/IEC
Short-Circuit Complete
Short-Circuit ANSI
Short-Circuit IEC 61363
Short-Circuit DC
Simulation RMS
Simulation EMT
Arc-Flash Analysis
Power Quality/Harmonics
Tie Open Point Opt.
Reliability
Hosting Capacity Analysis
Power Park Energy Analysis
Optimal Power Flow
Unit Commitment
Optimal Equipment Placement
Name
Bus1
Type
Equipment Type Library\Bus1
Zone
Area
Out of Service
System Type
AC
Usage
Busbar
Phase Technology
ABC
Nominal Voltage
Line-Line
132,
kV
Line-Ground
76,21024 kV
Earthed
OK
Cancel
Jump to ...
Cubicles



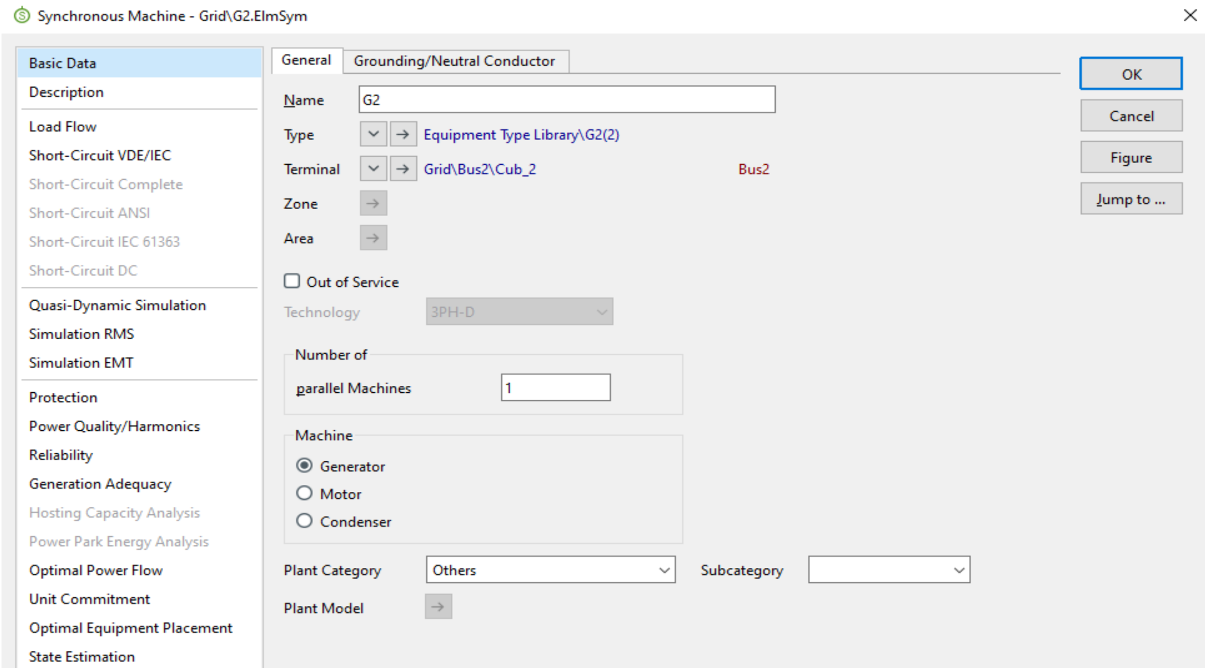
2. Configuring Bus 2

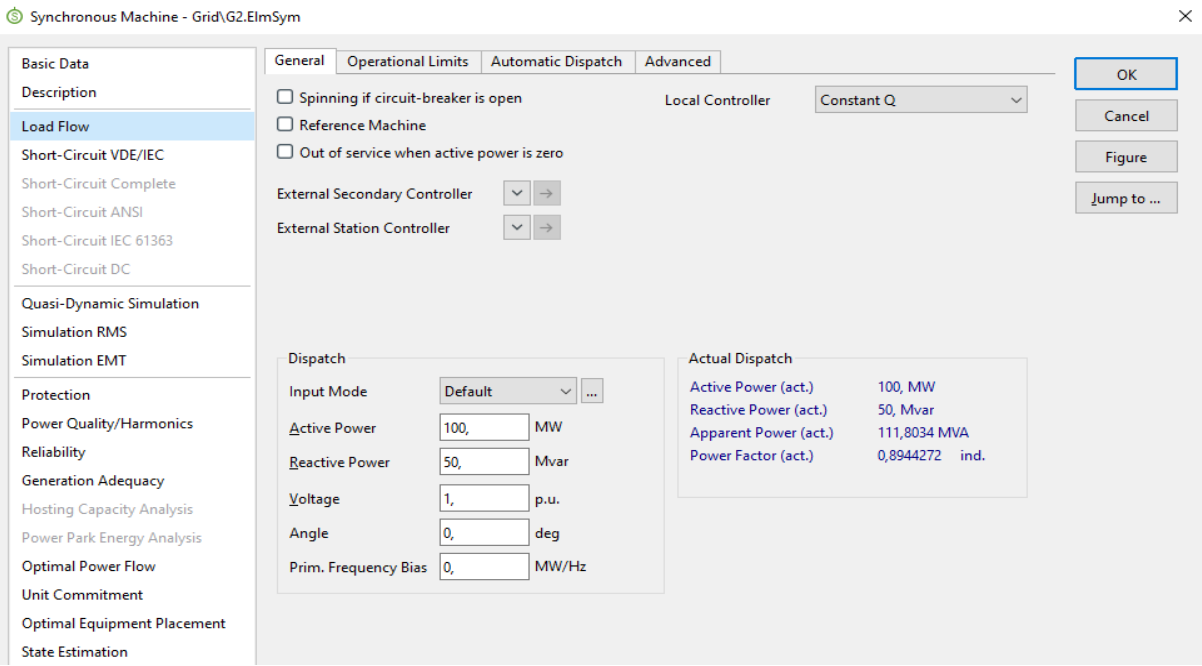

Terminal - Grid\Bus1.ElmTerm
Basic Data
Description
Load Flow
Short-Circuit VDE/IEC
Short-Circuit Complete
Short-Circuit ANSI
Short-Circuit IEC 61363
Short-Circuit DC
Simulation RMS
Simulation EMT
Arc-Flash Analysis
Power Quality/Harmonics
Tie Open Point Opt.
Reliability
Hosting Capacity Analysis
Power Park Energy Analysis
Optimal Power Flow
Unit Commitment
Optimal Equipment Placement
Voltage Control
Target Voltage 1, p.u. 132, kV
Delta V max 5, %
Delta V min -5, %
Priority -1
Steady State Voltage Limits
Upper Voltage Limit 1,05 p.u.
Lower Voltage Limit 0, p.u.
Rated Current 0,72 kA
OK
Cancel
Jump to ...
Cubicles

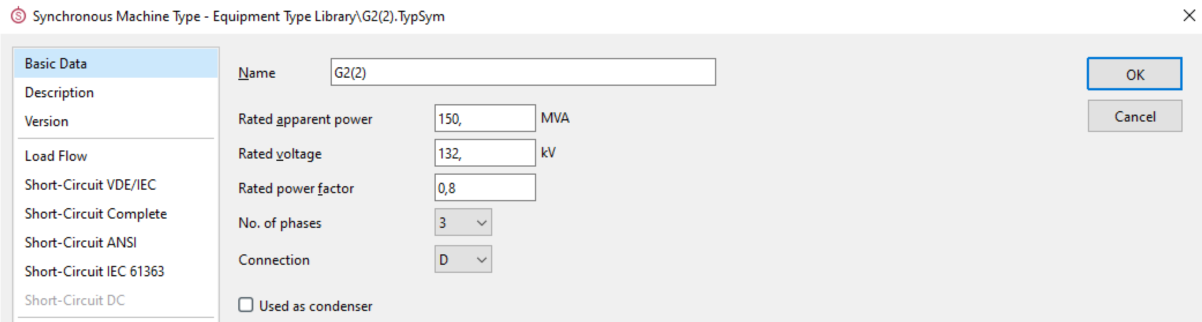

Terminal - Grid\Bus1.ElmTerm
Basic Data
Description
Load Flow
Short-Circuit VDE/IEC
Short-Circuit Complete
Short-Circuit ANSI
Short-Circuit IEC 61363
Short-Circuit DC
Simulation RMS
Simulation EMT
Arc-Flash Analysis
Power Quality/Harmonics
Tie Open Point Opt.
Reliability
Hosting Capacity Analysis
Power Park Energy Analysis
Optimal Power Flow
Unit Commitment
Optimal Equipment Placement
Name Bus1
Type Equipment Type Library\Bus1
Zone
Area
Out of Service
System Type AC
Usage Busbar
Phase Technology ABC
Nominal Voltage
Line-Line 132, kV
Line-Ground 76,21024 kV
Earthed
OK
Cancel
Jump to ...
Cubicles



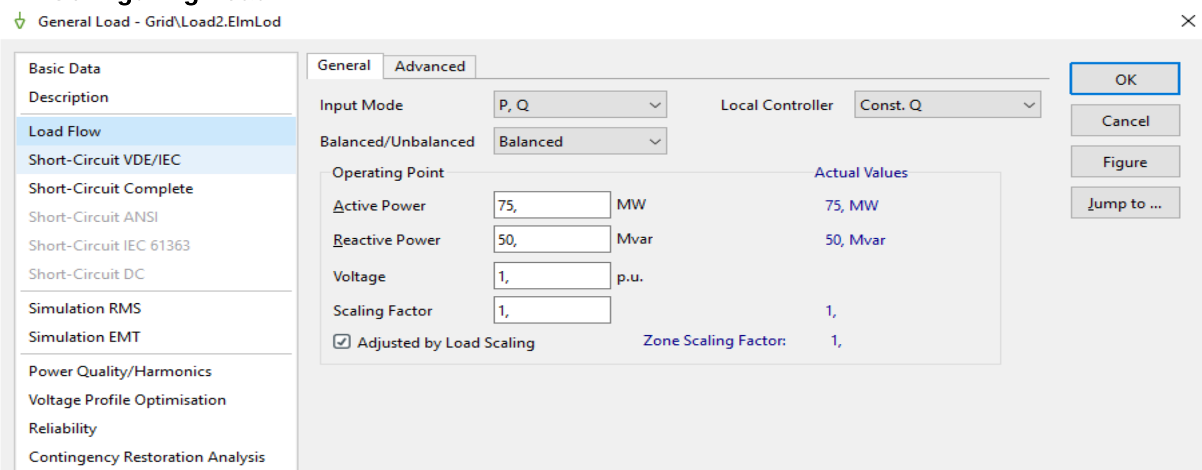
3. Configuring G2 (On bus 2)

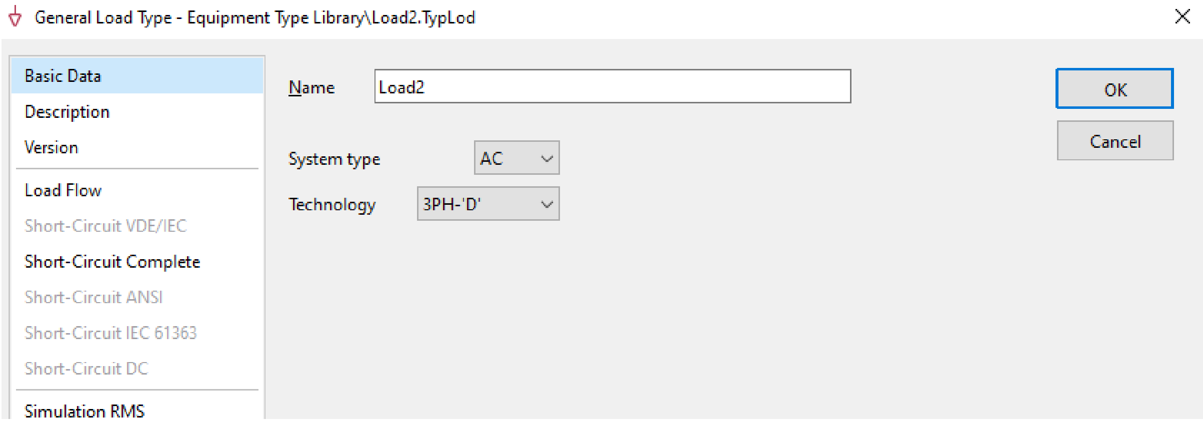




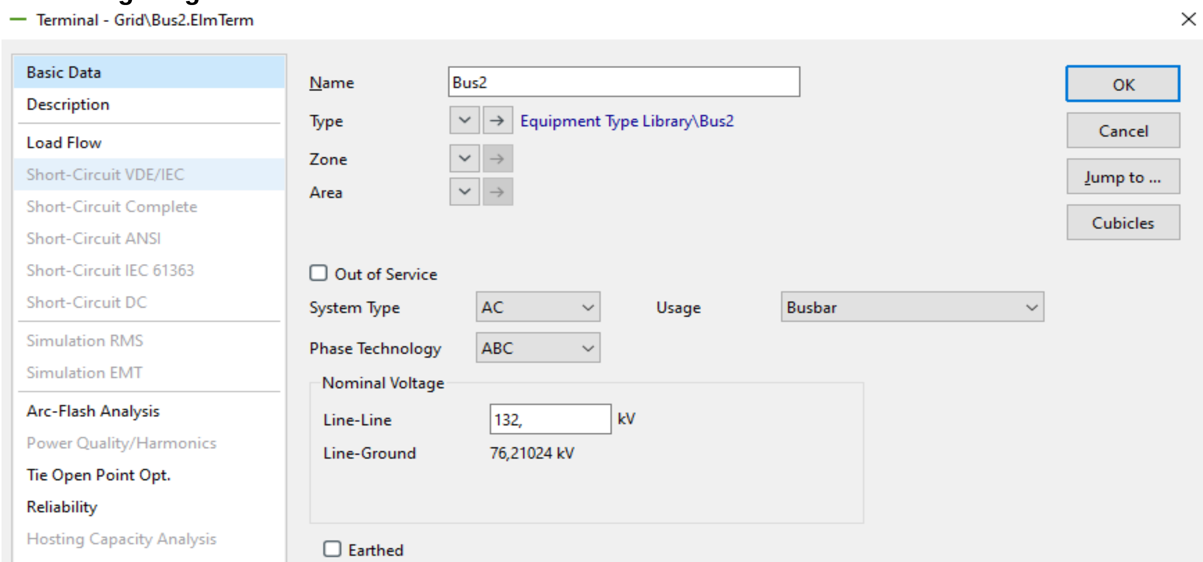


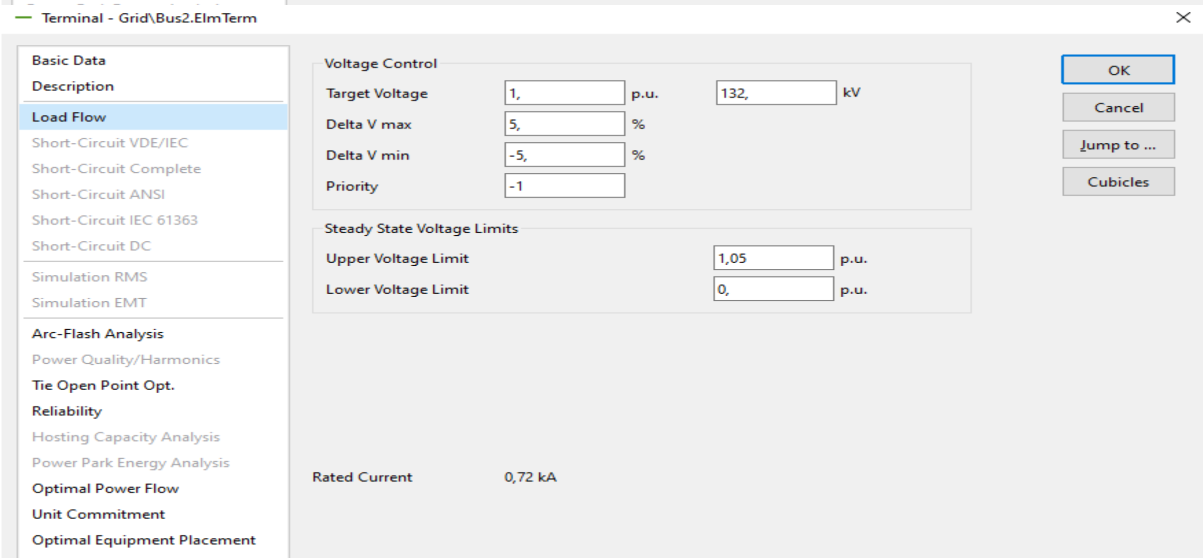
4. Configuring Load 2

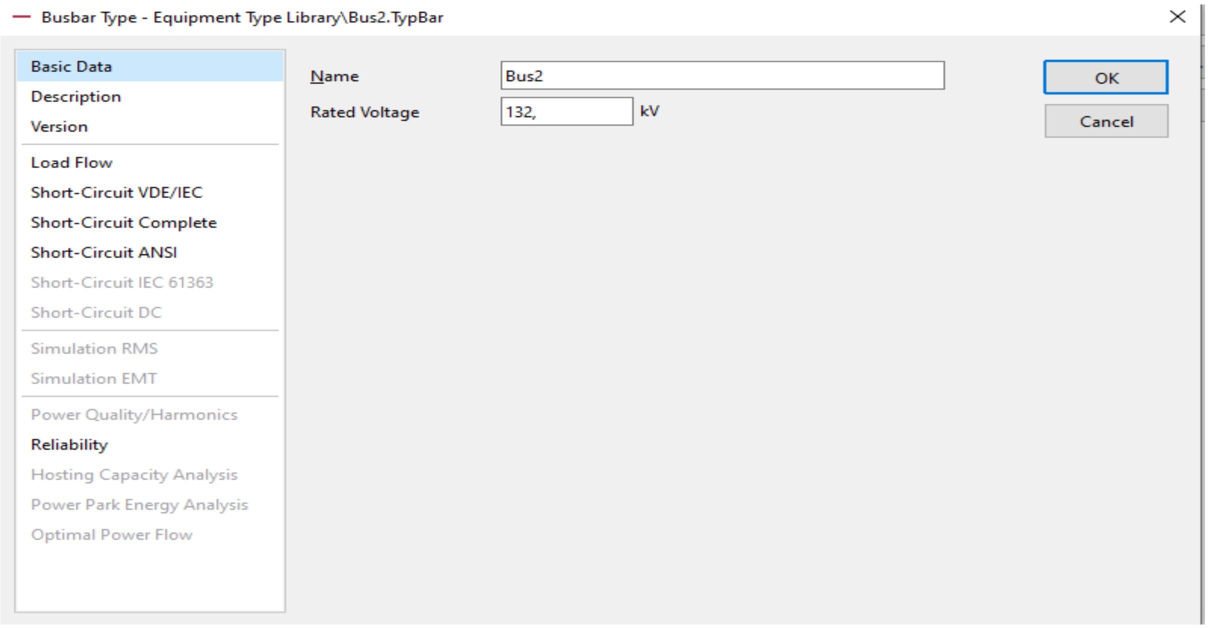


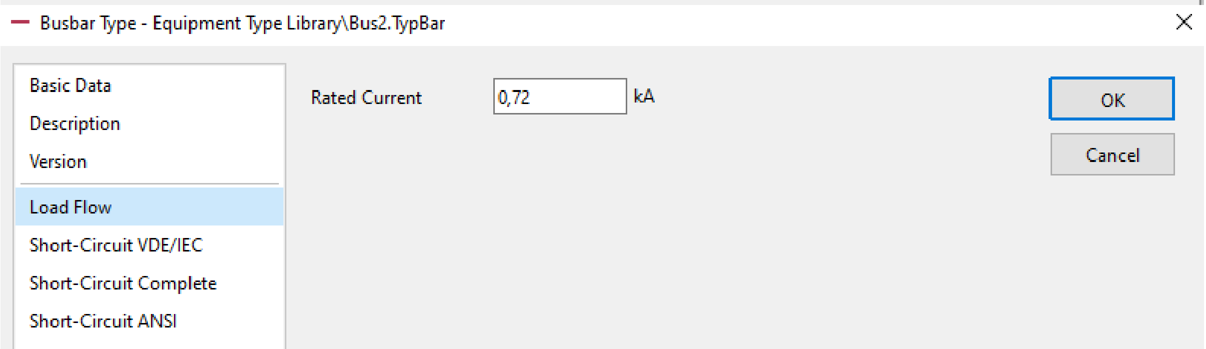


5. Configuring Bus 2

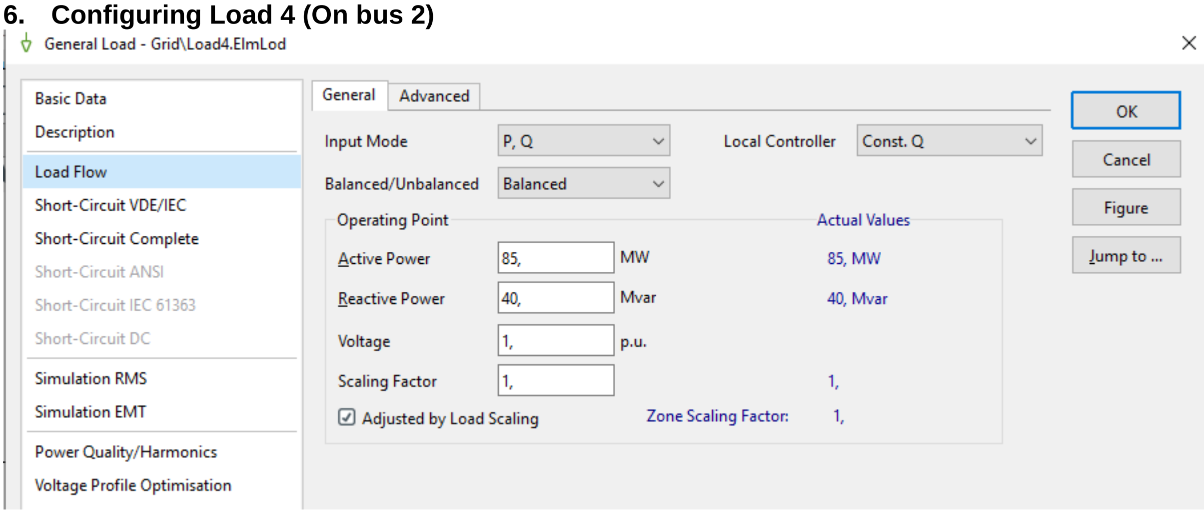




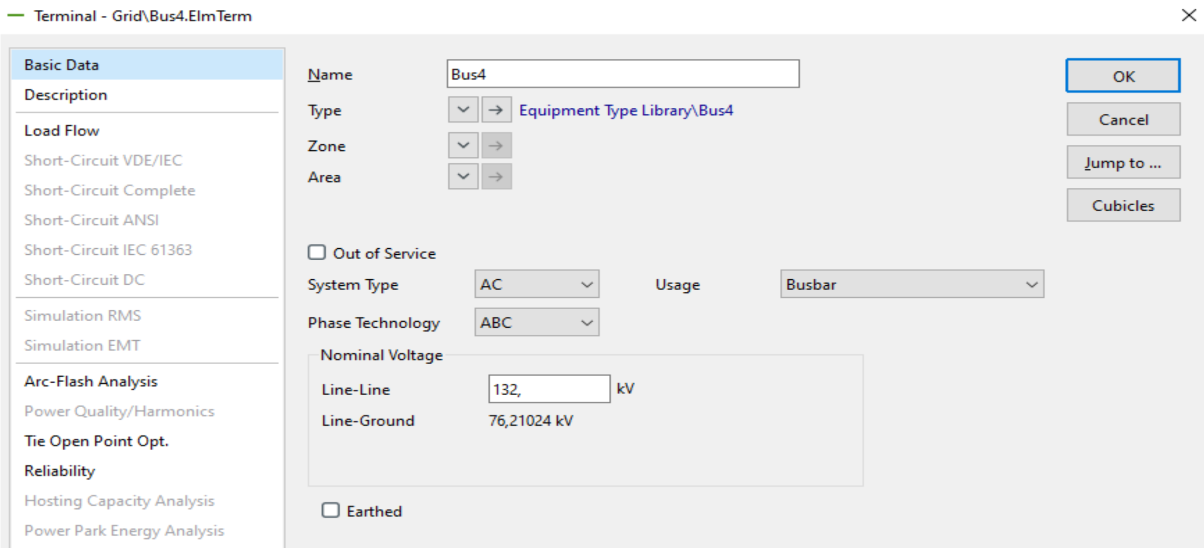


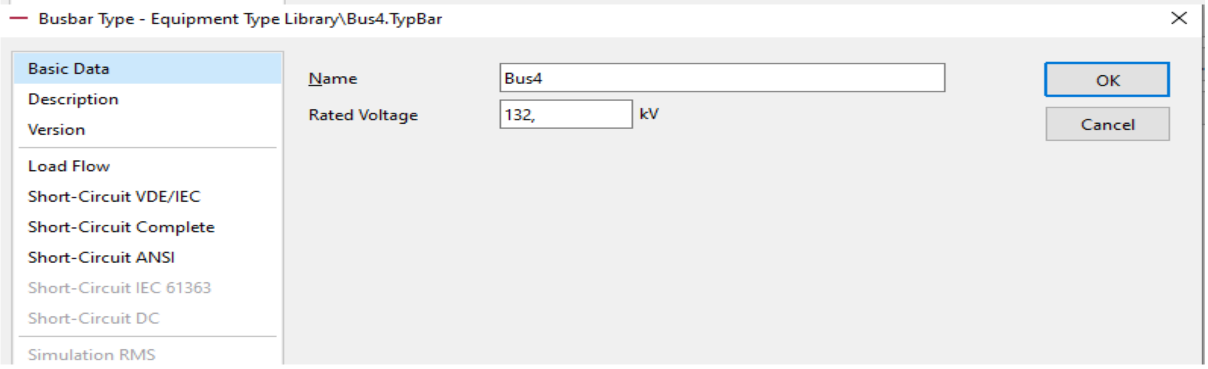


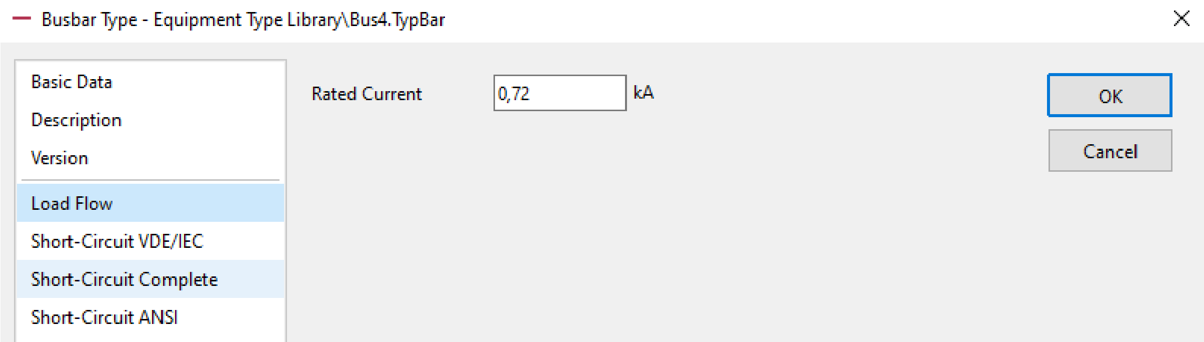
6. Configuring Load 4 (On bus 2)



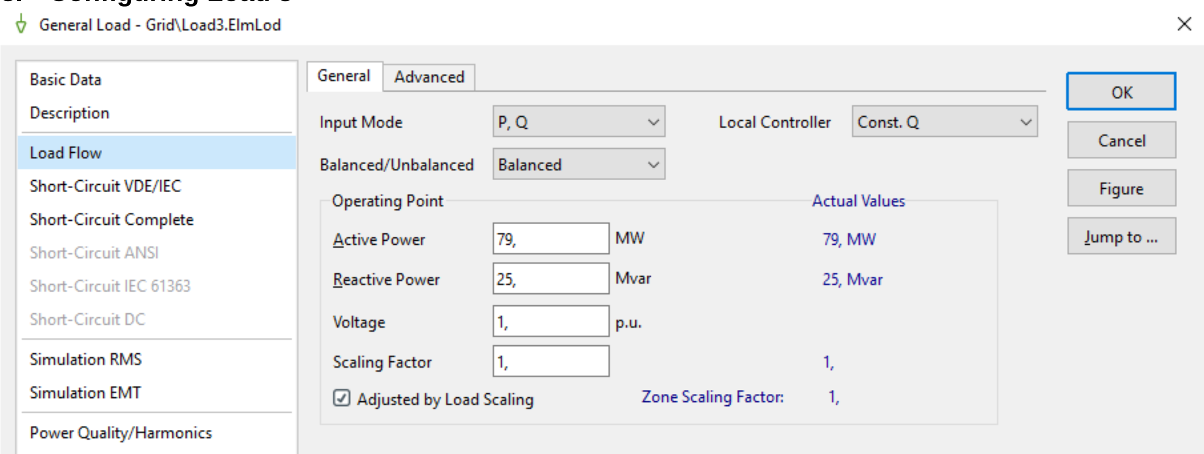
7. Configuring Bus 4



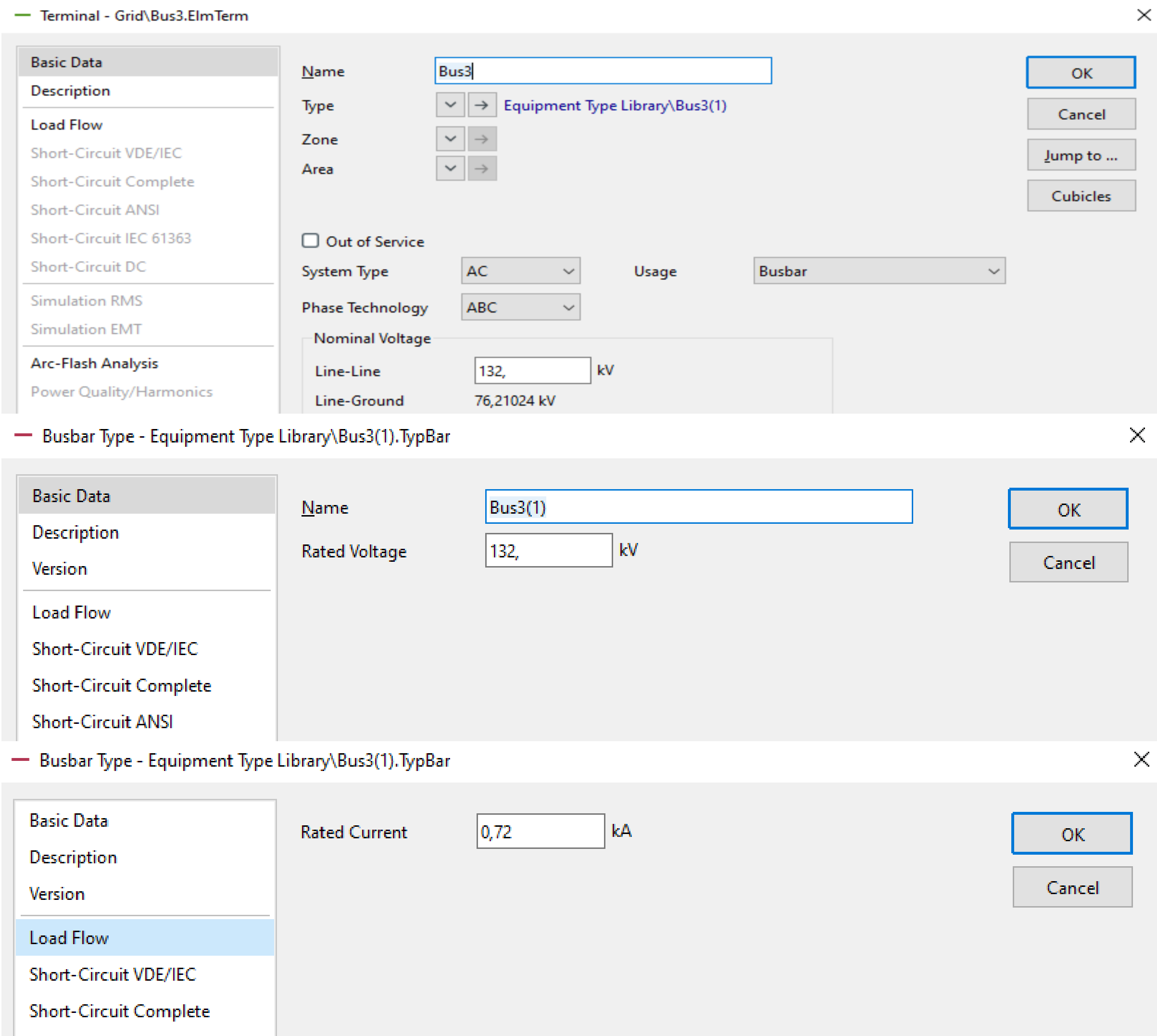




8. Configuring Load 3



9. Configuring Bus 3



Section B: Configuring Transmission lines

In load flow studies, the terms "rated current" and "line current" refer to different aspects of a power system's operation:

- The **rated current** of a component, such as a transmission line or a transformer, refers to its maximum designed operating current. It's the current level at which the component can operate continuously without overheating or suffering damage. Rated current is typically specified by the manufacturer and is used to determine the component's capacity and thermal limits.
- The **line current**, on the other hand, refers to the actual current flowing through a transmission line or a component during operation. In the context of a slack bus (also known as the swing bus or reference bus), the line current would refer to the current flowing into or out of the slack bus.
- DIgSILENT Slack Bus Configuration:
- If line current = 656 A, set rated current ≥ line current.
- **Reason**: Rated current should be above the expected operating current to avoid limiting performance and to handle balancing requirements.
- **Typical Practice**: Set rated current **10–20% higher** than expected line current to provide a safety margin.

Configuring Line Parameters in DIgSILENT from Per Unit Data

1. **Know the Base Values:**
    - Base power S_{base}
    - Base voltage V_{base}
2. **Find Base Impedance:**

$$Z_{\text{base}} = \frac{V_{\text{base}}^2}{S_{\text{base}}}$$

3. **Convert p.u. to Ohms:**

$$Z_{\text{ohm}} = Z_{\text{pu}} \times Z_{\text{base}}$$

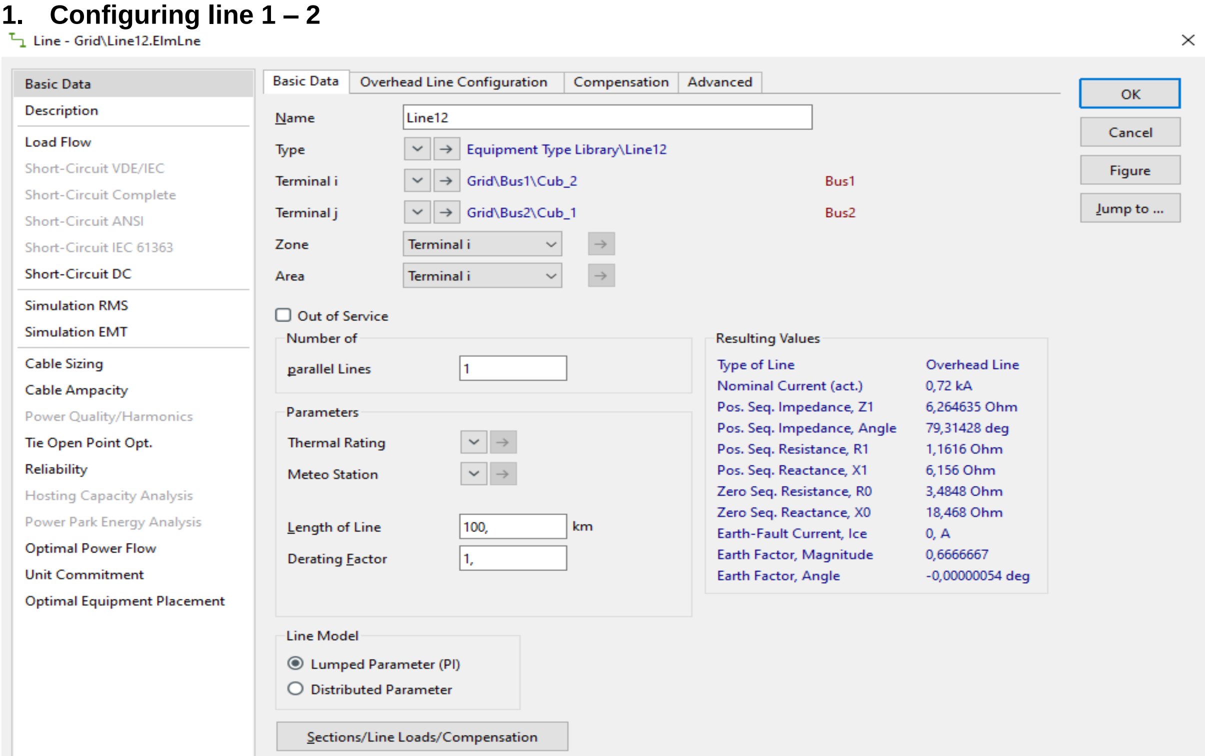
4. **Get Per-Length Values** (for DIgSILENT input):

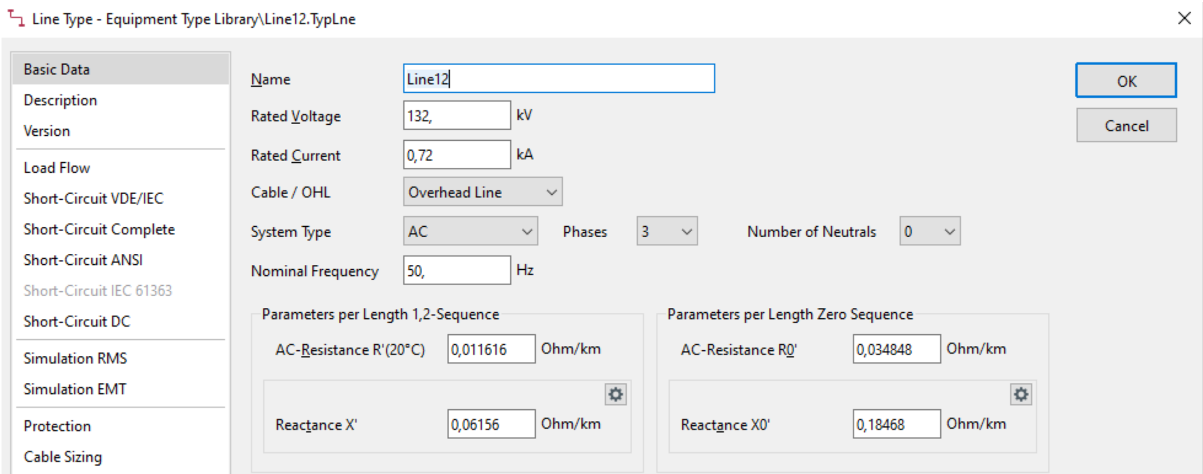
$$R_{\text{per km}} = \frac{R_{\text{ohm}}}{\text{Line length}}$$

$$X_{\text{per km}} = \frac{X_{\text{ohm}}}{\text{Line length}}$$

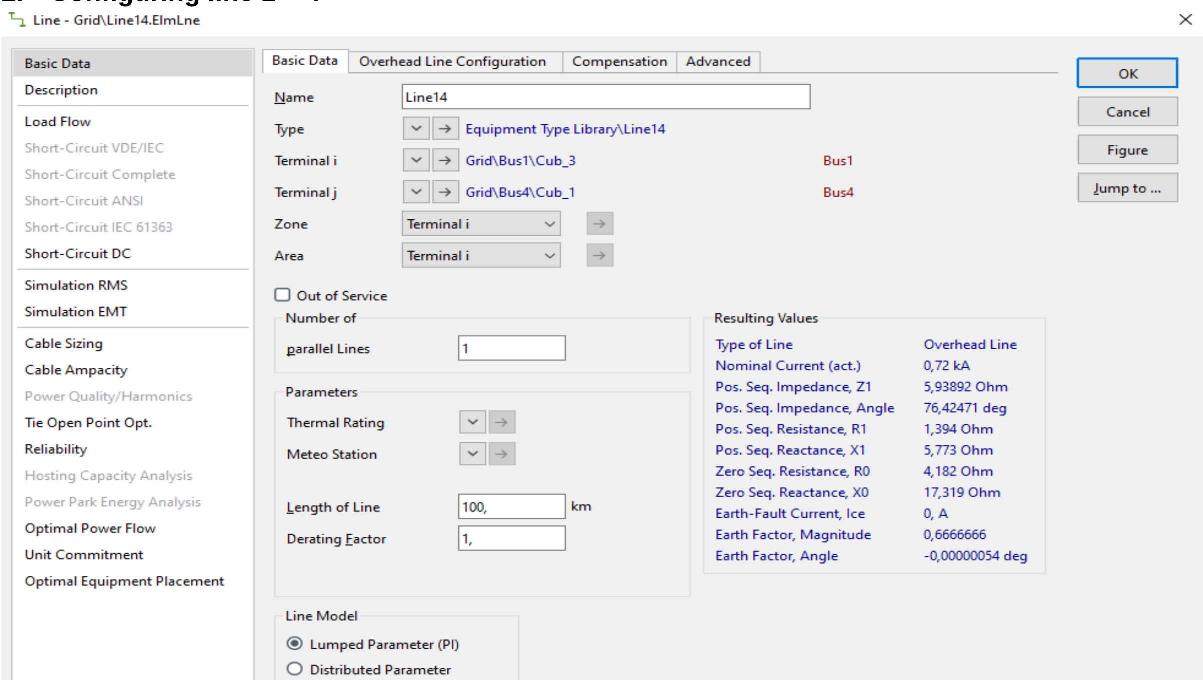
Then when you finish computing the per length 1,2 values, you configure the parameters per length zero sequence, if you are not given their values you multiply the per length 1,2 values by 3 to find their corresponding parameters per length zero sequence

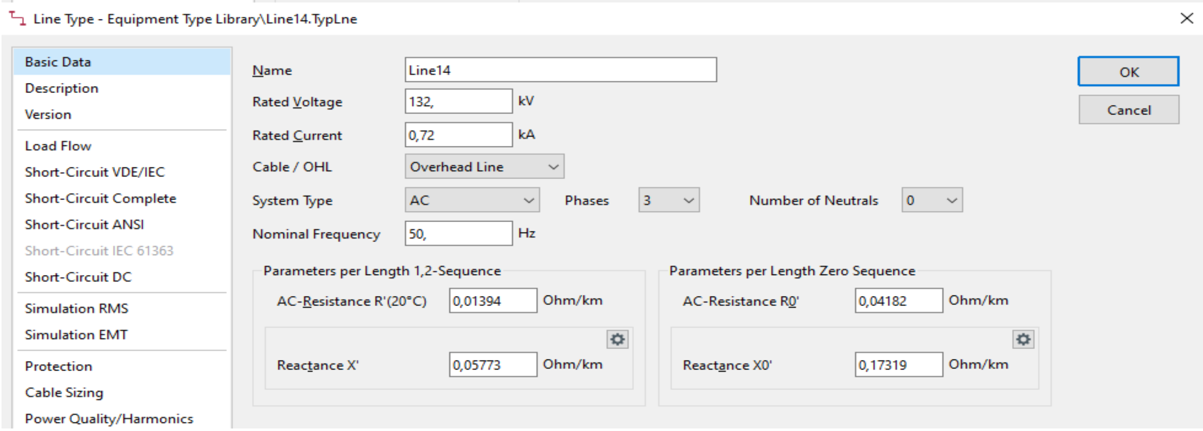
1. Configuring line 1 – 2



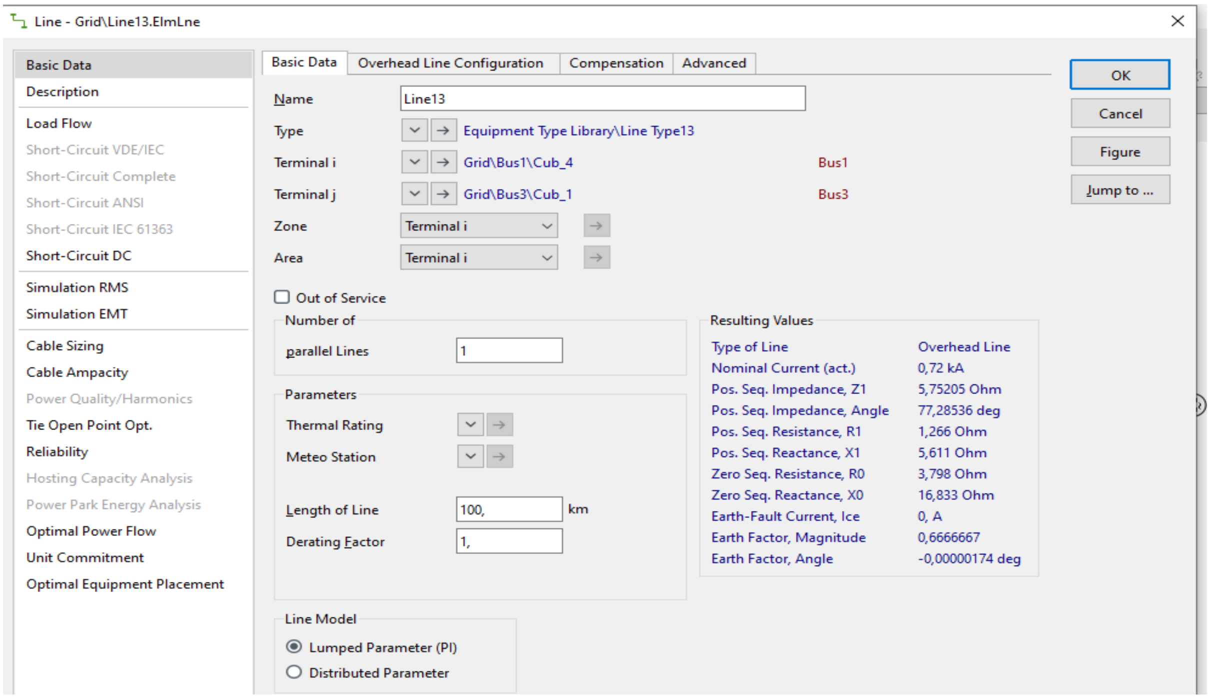


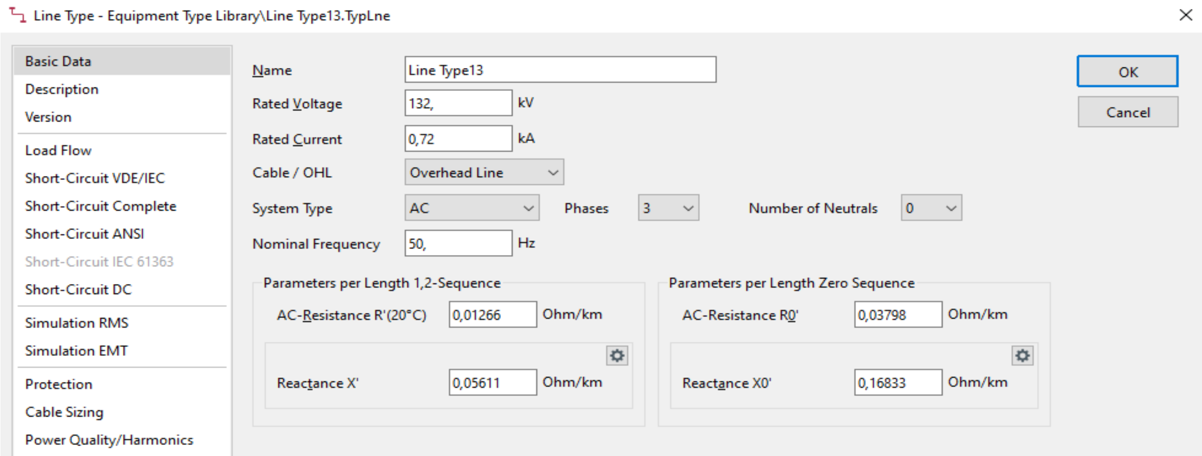
2. Configuring line 1 – 4



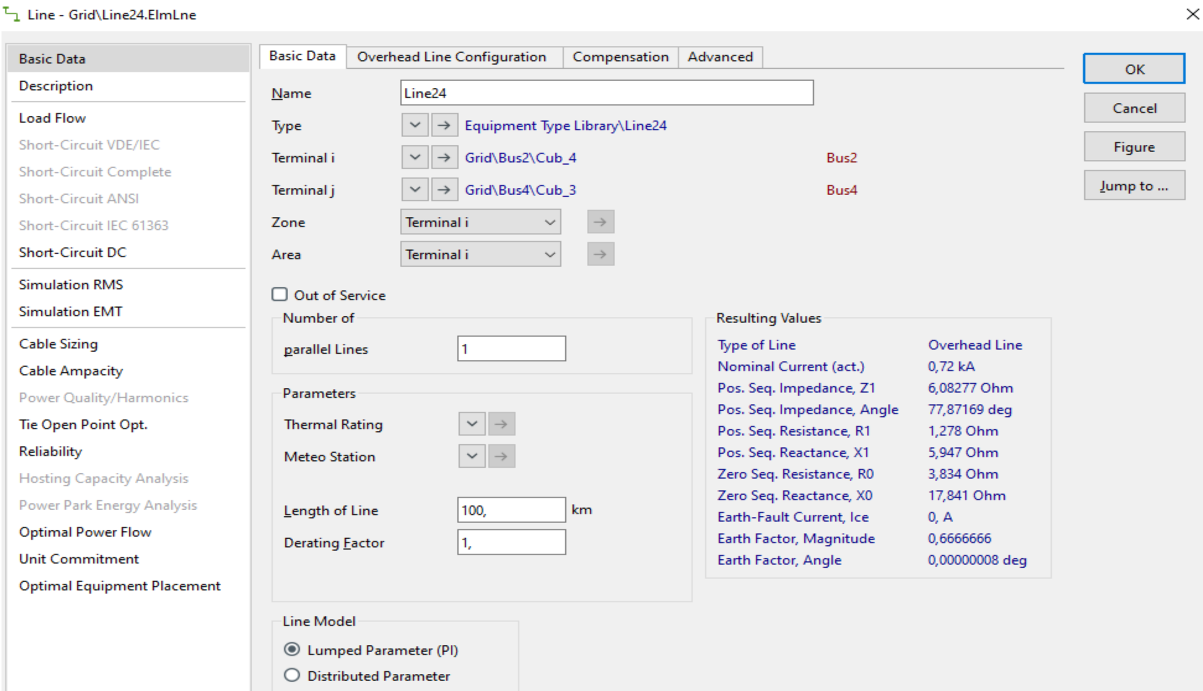


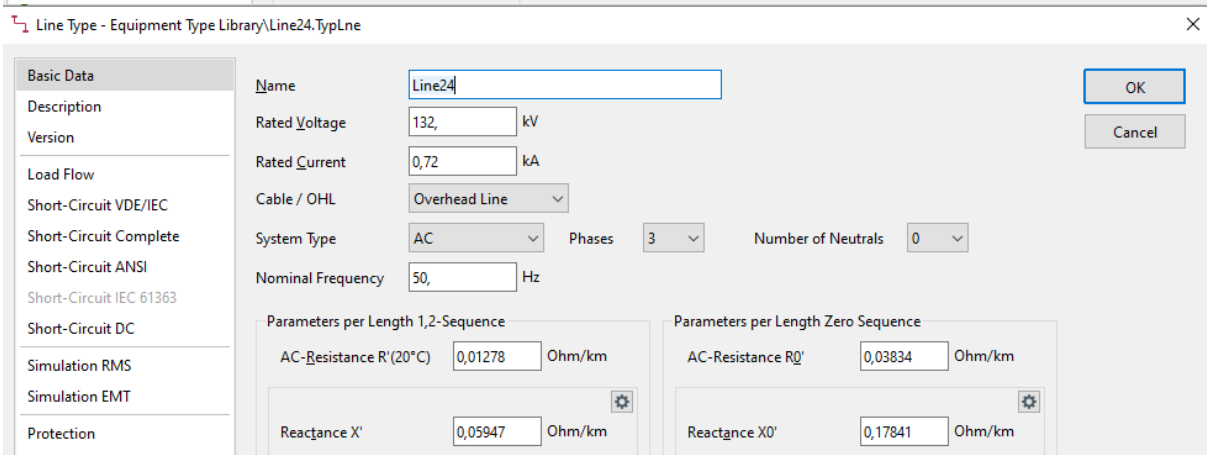
3. Configuring Line 1 – 3



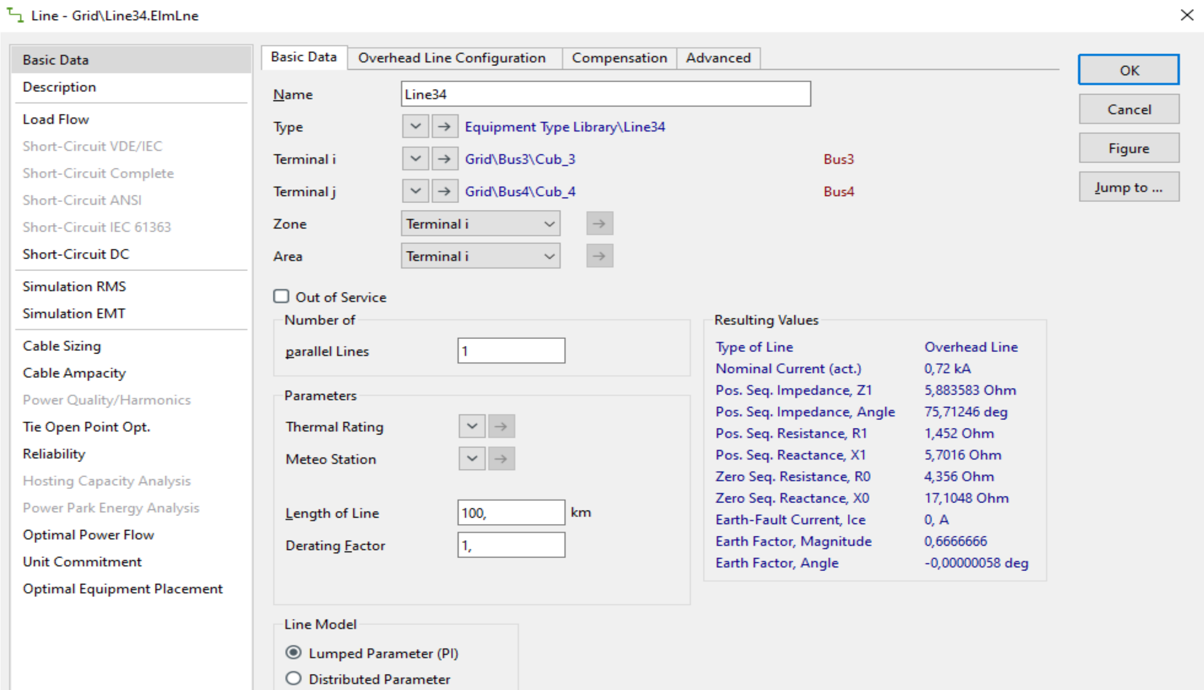


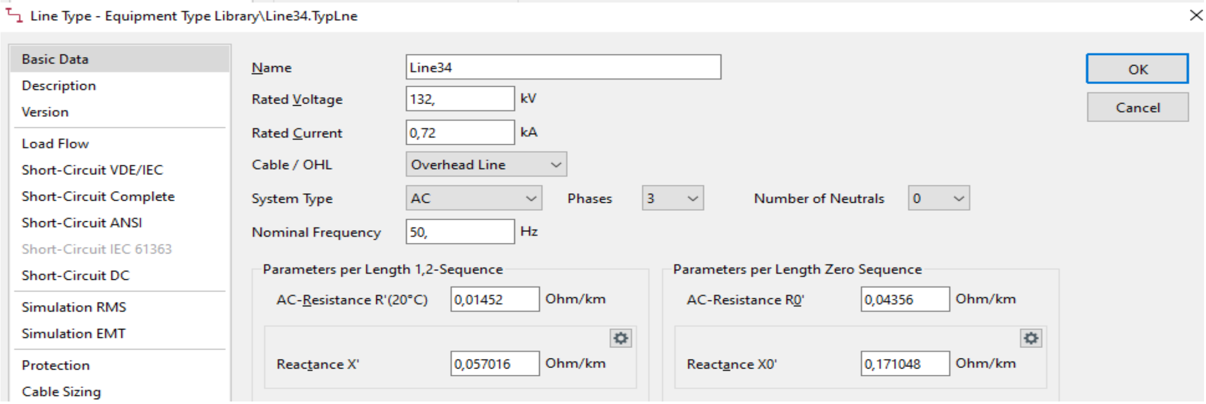
4. Configuring Line 2 – 4





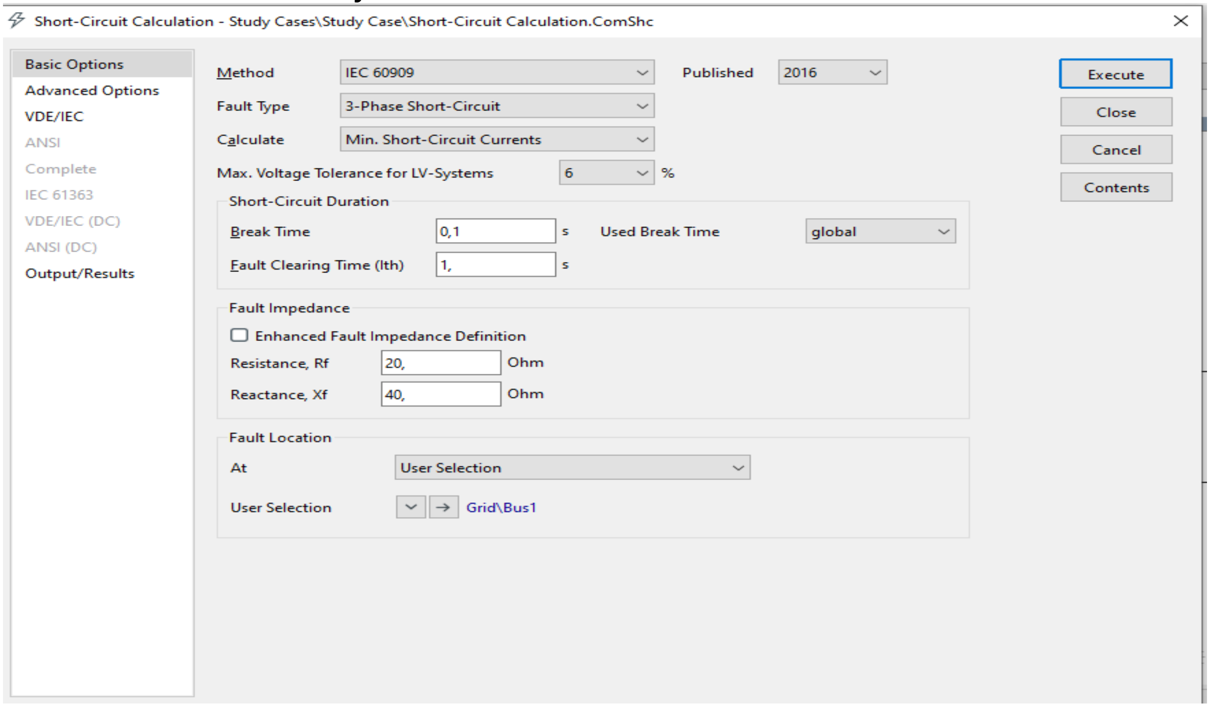
5. Configuring Line 3 – 4

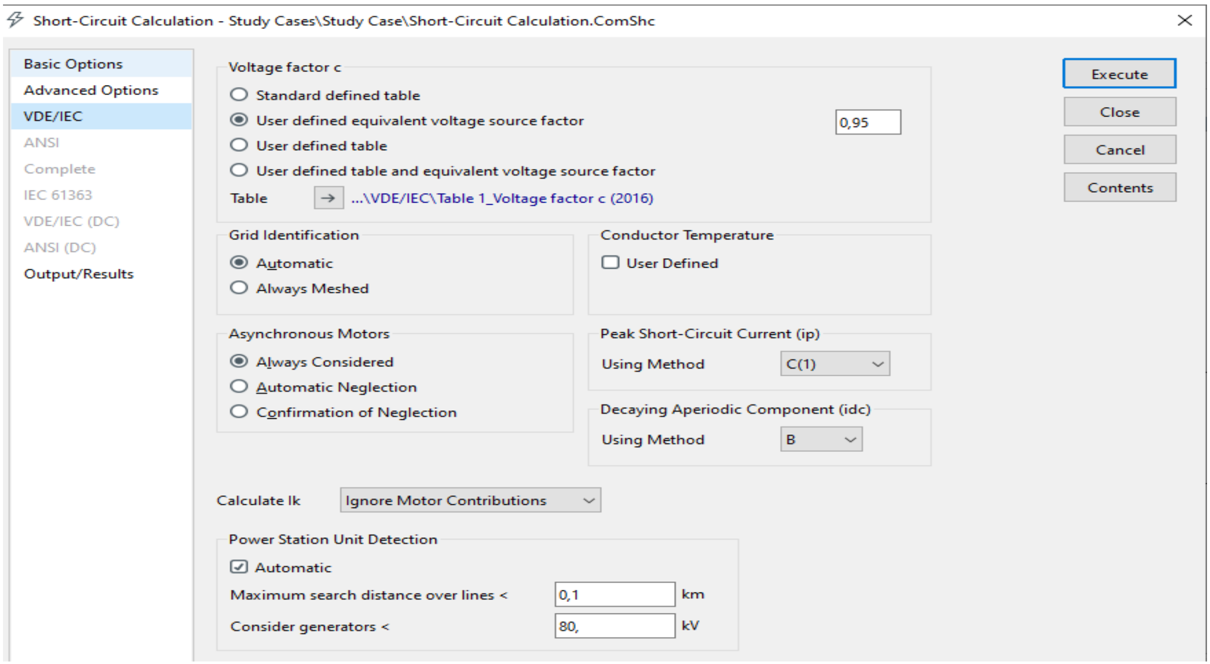




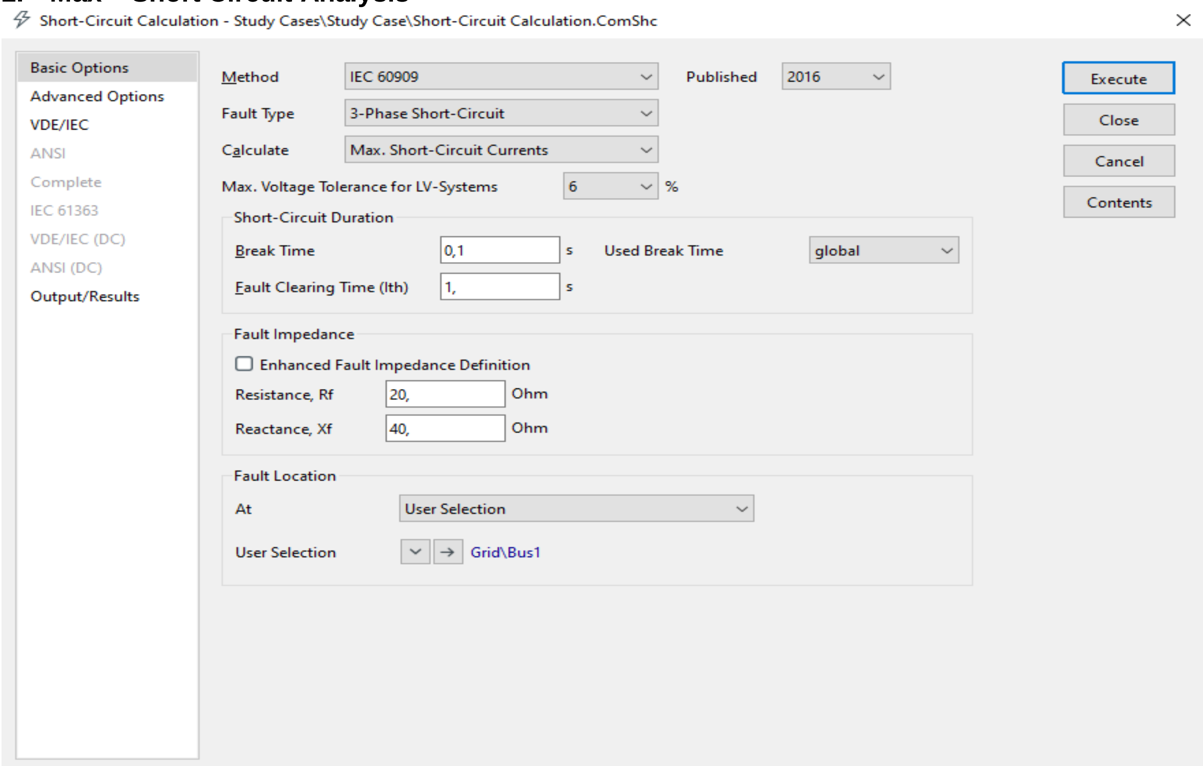
Short Circuit Analysis

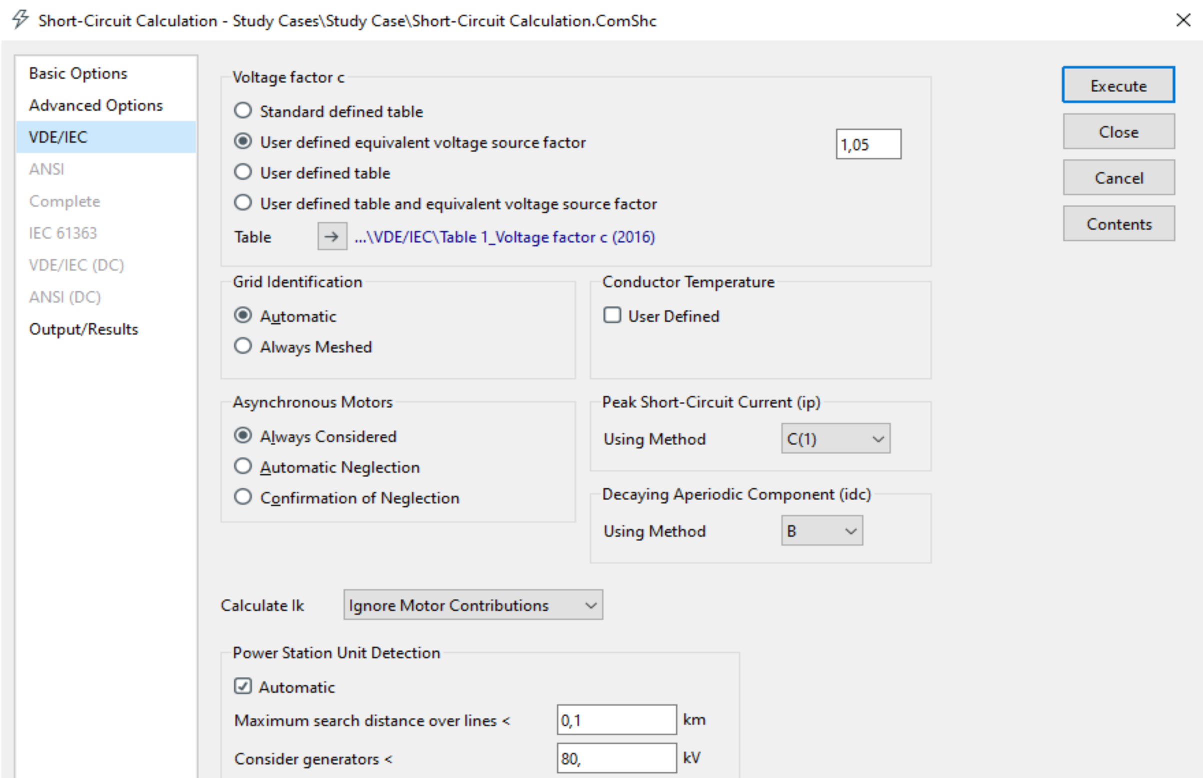
1. Min – Short Circuit Analysis





2. Max – Short Circuit Analysis





Short-Circuit Calculation - Study Cases\Study Case\Short-Circuit Calculation.ComShc
Basic Options
Advanced Options
VDE/IEC
ANSI
Complete
IEC 61363
VDE/IEC (DC)
ANSI (DC)
Output/Results
Voltage factor c
Standard defined table
User defined equivalent voltage source factor
1,05
User defined table
User defined table and equivalent voltage source factor
Table
...\VDE/IEC\Table 1_Voltage factor c (2016)
Grid Identification
Automatic
Always Meshed
Conductor Temperature
User Defined
Asynchronous Motors
Always Considered
Automatic Neglection
Confirmation of Neglection
Peak Short-Circuit Current (ip)
Using Method
C(1)
Decaying Aperiodic Component (idc)
Using Method
B
Calculate Ik
Ignore Motor Contributions
Power Station Unit Detection
Automatic
Maximum search distance over lines <
0,1
km
Consider generators <
80,
kV
Execute
Close
Cancel
Contents