

Power System Load Flow Analysis using an Excel Workbook

Abstract

The paper describes an MS-Excel Workbook, which illustrates four methods of power system load flow analysis. Iterative techniques are represented by the Newton-Raphson and Gauss-Seidel methods. The Workbook also includes two search algorithms: genetic algorithms and simulated annealing.

1. Introduction

Load flow studies are used to ensure that electrical power transfer from generators to consumers through the grid system is stable, reliable and economic. Conventional techniques for solving the load flow problem are iterative, using the Newton-Raphson or the Gauss-Seidel methods. Recently, however, there has been much interest in the application of stochastic search methods, such as Genetic Algorithms, to solving power system problems. The increasing presence of distributed alternative energy sources, often in geographically remote locations, complicates load flow studies and has triggered a resurgence of interest in the topic.

The principles of power system load flow studies are taught within elective modules in the later years of undergraduate electrical engineering courses, or as essential components of specialist masters programmes in electrical power engineering. From the educational viewpoint, therefore, the topic is important, yet a complete coverage presents some significant challenges. Pre-requisites include fundamental concepts from a.c. circuit analysis, such as phasor notation, impedance and admittance, power and reactive power, three-phase and per-unit systems, all of which are regarded as 'difficult' by many students. The load flow solution techniques bring extra mathematical hurdles, including matrix representation (with complex number coefficients), iterative methods and probability functions.

To assist in the teaching of load flow analysis techniques, the author developed an Excel Workbook in which four different methods were used to solve a simple load flow problem, which nevertheless has all of the features to be found in a larger-scale system. The Workbook allows students to vary both the problem being solved, by adjusting the power or voltage levels and line impedances, and also the parameters of the solution methods, such as numerical acceleration factors. This paper presents an overview of the general power system load flow problem and describes its

solution using four techniques: Newton-Raphson, Gauss-Seidel, Genetic Algorithm and Simulated Annealing. Also included are illustrative numerical results relating to the particular power system configuration analysed in the Workbook.

2. Formulation of the Load Flow Problem

Load flow studies are based on a nodal voltage analysis of a power system. As an example, consider the very simple system represented by the single-line diagram in Fig. 1. Here two generators (1 and 2) are interconnected by one transmission line and are separately connected to a load (3) by two other lines. If the phasor currents injected into the system are I_1 , I_2 , and I_3 , and the lines are modelled by simple series admittances, then it is possible to draw the equivalent circuit for one representative phase of the balanced three-phase system, as shown in Fig. 2.

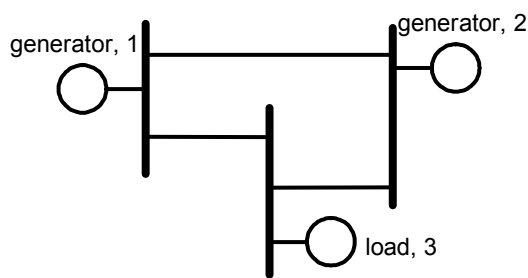


Fig. 1 Single-line diagram of a simple example power system

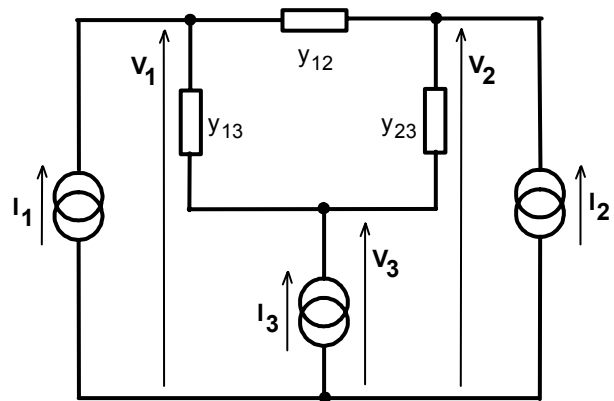


Fig. 2 Equivalent circuit for one phase of the system shown in Fig. 1

For the circuit in Fig. 2 the nodal voltage equations can be written directly. For example, at node 1:

$$I_1 = (y_{12} + y_{13})V_1 - y_{12}V_2 - y_{13}V_3 \quad (1)$$

In general, for a system with r nodes, then at node n :

$$I_n = Y_{n1}V_1 + Y_{n2}V_2 + \dots + Y_{nn}V_n + \dots + Y_{nr}V_r = \sum_{k=1}^r Y_{nk}V_k \quad (2)$$

where: Y_{nn} = sum of all admittances connected to node n

Y_{nk} = - (sum of all admittances connected between nodes n and k) = Y_{kn}

I_n = current injected at node n

For the complete system of r nodes:

$$\begin{bmatrix} I_1 \\ \vdots \\ I_n \\ \vdots \\ I_r \end{bmatrix} = \begin{bmatrix} Y_{11} & \dots & Y_{1n} & \dots & Y_{1r} \\ \vdots & & \vdots & & \vdots \\ Y_{n1} & \dots & Y_{nn} & \dots & Y_{nr} \\ \vdots & & \vdots & & \vdots \\ Y_{r1} & \dots & Y_{rn} & \dots & Y_{rr} \end{bmatrix} \begin{bmatrix} V_1 \\ \vdots \\ V_n \\ \vdots \\ V_r \end{bmatrix} \quad \text{or} \quad [I] = [Y][V] \quad (3)$$

where $[Y]$ is the nodal admittance matrix. Formulation of the load flow problem is most conveniently carried out with the terms in the nodal admittance matrix expressed in polar notation: $Y_{kn} = |Y_{kn}| \angle \theta_{kn}$. The Excel Workbook (Sheet 2) allows users to enter series impedance data for the three lines and then automatically calculates the terms in the nodal admittance matrix.

Conventional circuit analysis proceeds directly from equation (3) by inverting the nodal admittance matrix and hence solving for the nodal voltages $[V]$. However, the load flow problem is complicated by the lack of uniformity in the data about electrical conditions at the nodes. There are three distinct types of nodal data, which relate to the physical nature of the power system:

- a) Load nodes, where complex power $S_{ns} = P_{ns} + jQ_{ns}$ taken from or injected into the system is defined. Such nodes may also include links to other systems. At these load nodes, the voltage magnitude $|V_n|$ and phase angle δ_n must be calculated.
- b) Generator nodes, where the injected power, P_{ns} , and the magnitude of the nodal voltage $|V_n|$ are specified. These constraints reflect the generator's operating characteristics, in which power is controlled by the governor and terminal voltage is controlled by the automatic voltage regulator. At the generator nodes the voltage phase angle δ_n must be calculated
- c) At least one node, termed the 'floating bus' or 'slack bus', where the nodal voltage magnitude $|V_n|$ and phase angle δ_n are specified. This node acts as the reference node and is commonly chosen to have a phase angle $\delta_n = 0^\circ$. The power and reactive power delivered at this node are not specified.

In the system configuration of Fig. 1, which is analysed in the Excel Workbook, each type of node is represented with node 1 being a floating bus, node 2 being a generator node and node 3 being a load node. Consequently values must be specified for the power (P_{2s}) injected at node 2, and the power (P_{3s}) and reactive

power (Q_{3s}) injected at node 3. All three of these power values may be changed by the user, though default values are provided ($P_{2s} = 1.0$; $P_{3s} = -1.5$; $Q_{3s} = -0.2$), with negative values indicating that power or reactive power is being drawn from the system. The magnitude of the voltage at node 1 can be specified, with the default value being 1.0 pu, while the phase angle is fixed at 0° ($\mathbf{V}_1 = 1.0 \angle 0^\circ$). At the generator node (node 2), the voltage magnitude can be set by the user with the default value being 1.1 pu ($\mathbf{V}_2 = 1.1 \angle \delta_2$), and the phase angle δ_2 is calculated during the load flow solution. At the load node (node 3) the voltage magnitude and phase angle have to be calculated ($\mathbf{V}_3 = |V_3| \angle \delta_3$). So the complete load flow problem for this particular power system configuration involves the calculation of the voltage magnitude $|V_3|$ and the phase angles δ_2, δ_3 .

3. Newton Raphson Method

3.1. General Approach

The Newton-Raphson method is an iterative technique for solving systems of simultaneous equations in the general form:

$$\begin{aligned} f_1(x_1, \dots, x_n, \dots, x_r) &= K_1 \\ f_j(x_1, \dots, x_n, \dots, x_r) &= K_n \\ f_n(x_1, \dots, x_n, \dots, x_r) &= K_r \end{aligned} \tag{4}$$

where $f_1, \dots, f_n, \dots, f_r$ are differentiable functions of the variables $x_1, \dots, x_n, \dots, x_r$ and $K_1, \dots, K_n, \dots, K_r$ are constants. Applied to the load flow problem, the variables are the nodal voltage magnitudes and phase angles, the functions are the relationships between power, reactive power and node voltages, while the constants are the specified values of power and reactive power at the generator and load nodes.

Power and reactive power functions can be derived by starting from the general expression for injected current (Eqn. 2) at node n:

$$I_n = \sum_{k=1}^r Y_{nk} V_k$$

so the complex power input to the system at node n is:

$$\mathbf{S}_n = \mathbf{V}_n \mathbf{I}_n^* \tag{5}$$

where the superscript * denotes the complex conjugate. Substituting from (2) with all complex variables written in polar form:

$$\mathbf{S}_n = \mathbf{V}_n \sum_{k=1}^r Y_{nk}^* \mathbf{V}_k^* = \sum_{k=1}^r |V_n| |V_k| |Y_{nk}| \angle \{\delta_n - \delta_k - \theta_{nk}\} \quad (6)$$

The power and reactive power inputs at node n are derived by taking the real and imaginary parts of the complex power:

$$P_n = \Re\{\mathbf{S}_n\} = \sum_{k=1}^r |V_n| |V_k| |Y_{nk}| \cos\{\delta_n - \delta_k - \theta_{nk}\} \quad (7)$$

$$Q_n = \Im\{\mathbf{S}_n\} = \sum_{k=1}^r |V_n| |V_k| |Y_{nk}| \sin\{\delta_n - \delta_k - \theta_{nk}\} \quad (8)$$

The load flow problem is to find values of voltage magnitude and phase angle, which, when substituted into (7) and (8), produce values of power and reactive power equal to the specified set values at that node, P_{ns} and Q_{ns} .

The first step in the solution is to make initial estimates of all the variables: $|V_n^0|, \delta_n^0$ where the superscript 0 indicates the number of iterative cycles completed. Using these estimates, the power and reactive power input at each node can be calculated from (7) and (8). These values are compared with the specified values to give a power and reactive power error. For node n:

$$\Delta P_n^0 = P_{ns} - \sum_{k=1}^r |V_n^0| |V_k^0| |Y_{nk}| \cos\{\delta_n^0 - \delta_k^0 - \theta_{nk}\} \quad (9)$$

$$\Delta Q_n^0 = Q_{ns} - \sum_{k=1}^r |V_n^0| |V_k^0| |Y_{nk}| \sin\{\delta_n^0 - \delta_k^0 - \theta_{nk}\} \quad (10)$$

The power and reactive power errors at each node are related to the errors in the voltage magnitudes and phase angles, e.g. $\Delta|V_n|^0, \Delta\delta_n^0$, by the first order approximations:

$$\begin{bmatrix} \vdots \\ \Delta P_n^0 \\ \vdots \\ \vdots \\ \Delta Q_n^0 \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \frac{\partial P_n}{\partial \delta_{n-1}} & \frac{\partial P_n}{\partial \delta_n} & \frac{\partial P_n}{\partial \delta_{n+1}} & \vdots & \vdots & \vdots & \frac{\partial P_n}{\partial |V_{n-1}|} & \frac{\partial P_n}{\partial |V_n|} & \frac{\partial P_n}{\partial |V_{n+1}|} & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \frac{\partial Q_n}{\partial \delta_{n-1}} & \frac{\partial Q_n}{\partial \delta_n} & \frac{\partial Q_n}{\partial \delta_{n+1}} & \vdots & \vdots & \vdots & \frac{\partial Q_n}{\partial |V_{n-1}|} & \frac{\partial Q_n}{\partial |V_n|} & \frac{\partial Q_n}{\partial |V_{n+1}|} & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \Delta \delta_{n-1}^0 \\ \Delta \delta_n^0 \\ \Delta \delta_{n+1}^0 \\ \vdots \\ \Delta |V_{n-1}|^0 \\ \Delta |V_n|^0 \\ \Delta |V_{n+1}|^0 \end{bmatrix} \quad (11)$$

where the matrix of partial differentials is called the Jacobian matrix, [J]. The elements of the Jacobian are calculated by differentiating the power and reactive power expressions (7,8) and substituting the estimated values of voltage magnitude and phase angle.

At the next stage of the Newton-Raphson solution, the Jacobian is inverted. Matrix inversion is a computationally-complex task with the resources of time and storage increasing rapidly with the order of [J]. This requirement for matrix inversion is a major drawback of the Newton-Raphson method of load flow analysis for large-scale power systems. However, with the inversion completed, the approximate errors in voltage magnitudes and phase angles can be calculated by pre-multiplying both sides of (11):

$$\begin{bmatrix} \Delta\delta_{n-1}^0 \\ \Delta\delta_n^0 \\ \Delta\delta_{n+1}^0 \\ \dots \\ \Delta|V_{n-1}|^0 \\ \Delta|V_n|^0 \\ \Delta|V_{n+1}|^0 \end{bmatrix} = \begin{bmatrix} \mathbf{J}^0 \end{bmatrix}^{-1} \begin{bmatrix} \vdots \\ \Delta P_n^0 \\ \vdots \\ \dots \\ \vdots \\ \Delta Q_n^0 \\ \vdots \end{bmatrix} \quad (12)$$

The approximate errors from (12) are added to the initial estimates to produce new estimated values of node voltage magnitude and angle. For node n:

$$|V_n^1| = |V_n^0| + \Delta|V_n^0| \quad (13) \quad \delta_n^1 = \delta_n^0 + \Delta\delta_n^0 \quad (14)$$

Because first-order approximations are used in (11) the new estimates (denoted by the superscript ¹) are not exact solutions to the problem. However, they can be used in another iterative cycle, involving the solution of Equations (9-14). The process is repeated until the differences between successive estimates are within an acceptable tolerance band.

The description above relates specifically to a load node, where there are two unknowns (the voltage magnitude and angle) and two equations relating to the specified power and reactive power. For a generator node the voltage magnitude $|V_n|$ and power P_n are specified, but the reactive power is not specified. The order of the calculation can be reduced by 1. There is no need to ensure that the reactive power is at a set value and only the angle of the node voltage needs to be calculated, so

one row and column are removed from the Jacobian. For the floating bus, both voltage magnitude and angle are specified, so there is no need to calculate these quantities.

3.2. Application of the Newton-Raphson Method to the Specific Problem

For the example system shown in Fig. 1 and analysed in the Excel Workbook, there are three unknowns ($|V_3|, \delta_2, \delta_3$) and three set values of power and reactive power (P_{2s}, P_{3s}, Q_{3s}). General expressions for power and reactive power input at nodes 2 and 3 can be derived from (7) and (8):

$$P_2 = |V_2||V_1||Y_{21}|\cos\{\delta_2 - \delta_1 - \theta_{21}\} + |V_2|^2|Y_{22}|\cos\{-\theta_{22}\} + |V_2||V_3||Y_{23}|\cos\{\delta_2 - \delta_3 - \theta_{23}\} \quad (15)$$

$$P_3 = |V_3||V_1||Y_{31}|\cos\{\delta_3 - \delta_1 - \theta_{31}\} + |V_3||V_2||Y_{32}|\cos\{\delta_3 - \delta_2 - \theta_{32}\} + |V_3|^2|Y_{33}|\cos\{-\theta_{33}\} \quad (16)$$

$$Q_3 = |V_3||V_1||Y_{31}|\sin\{\delta_3 - \delta_1 - \theta_{31}\} + |V_3||V_2||Y_{32}|\sin\{\delta_3 - \delta_2 - \theta_{32}\} + |V_3|^2|Y_{33}|\sin\{-\theta_{33}\} \quad (17)$$

In the iterative solution process, (15-17) are used to calculate the power and reactive power inputs from latest estimates of node voltages and then using (9,10) to calculate the power errors. The terms in the Jacobian are obtained by partial differentiation of (15-17). For example:

$$\frac{\partial P_3}{\partial |V_3|} = |V_1||Y_{31}|\cos\{\delta_3 - \delta_1 - \theta_{31}\} + |V_2||Y_{32}|\cos\{\delta_3 - \delta_2 - \theta_{32}\} + 2|V_3||Y_{33}|\cos\{-\theta_{33}\} \quad (18)$$

$$\frac{\partial P_3}{\partial \delta_2} = |V_3||V_2||Y_{32}|\sin\{\delta_3 - \delta_2 - \theta_{32}\} \quad (19)$$

Inversion of the 3x3 Jacobian with scalar elements is easily accomplished in Excel. (12) can be used to derive the approximate errors in the three variables and new estimates formed from (13,14). These new estimates are applied in the subsequent iterative cycle.

3.3. Sample Results for the Newton-Raphson Method

For the default system inputs, defined in Section 2, the Newton-Raphson method, implemented in the 3rd sheet of the Excel Workbook, produces the results shown in Table 1. In this simple example, only 5 iterations are needed to generate results which are stable to three decimal places, even though the initial estimates of all three unknowns are far from the correct value. When making comparisons with other methods of solution, it is important to realise that each iterative cycle of the Newton-

Raphson method involves considerable computational effort, notably to invert the Jacobian.

4. Gauss-Seidel Method

4.1. General Approach

The Gauss-Seidel Method is another iterative technique for solving the load flow problem, by successive estimation of the node voltages. Equation (2) can be re-arranged to give an expression for the complex conjugate of the current input at node n:

$$I_n^* = \sum_{k=1}^{n-1} Y_{nk}^* V_k^* + Y_{nn}^* V_n^* + \sum_{k=n+1}^r Y_{nk}^* V_k^* \quad (20)$$

Substituting for I_n^* from (20) into (5):

$$\frac{S_n}{V_n} = \sum_{k=1}^{n-1} Y_{nk}^* V_k^* + Y_{nn}^* V_n^* + \sum_{k=n+1}^r Y_{nk}^* V_k^* \quad (21)$$

and re-arranging:

$$V_n^* = - \sum_{k=1}^{n-1} \frac{Y_{nk}^* V_k^*}{Y_{nn}^*} - \sum_{k=n+1}^r \frac{Y_{nk}^* V_k^*}{Y_{nn}^*} + \frac{S_n}{V_n Y_{nn}^*} \quad (22)$$

The node voltage V_n appears on both sides of (22), which cannot, therefore, be used to give a direct solution. However, this equation is used in the Gauss-Seidel method as the basis for an iterative solution. If V_n^p and V_n^{p+1} denote the values of the voltage at node n after p and p+1 iteration cycles, (22) can be written:

$$V_n^{*(p+1)} = - \sum_{k=1}^{n-1} \frac{Y_{nk}^* V_k^{*(p+1)}}{Y_{nn}^*} - \sum_{k=n+1}^r \frac{Y_{nk}^* V_k^{*(p)}}{Y_{nn}^*} + \frac{S_n}{V_n^{(p)} Y_{nn}^*} \quad (23)$$

Note that in evaluating the nth node voltage, the latest estimates of the other node voltages are used. In the p+1th iteration cycle when performing the calculation for node n, the voltages at the nodes k=1...n-1 are available, but for the other node voltages the values from the previous (pth) cycle have to be used.

The foregoing discussion is appropriate to load nodes. At the floating bus the voltage V_n is known and so does not need to be calculated. Generator nodes are particularly problematical for the Gauss-Seidel Method. At these nodes the power P_n and voltage magnitude $|V_n|$ are specified, so in (23) V_n^{p+1} cannot be calculated because Q_n (the

imaginary part of \mathbf{S}_n) is unknown. This difficulty is addressed by first calculating Q_n , as follows:

$$\text{the voltage component: } (\mathbf{V}_n^{*(p+1)})^\# = -\sum_{k=1}^{n-1} \frac{Y_{nk}^* \mathbf{V}_k^{*(p+1)}}{Y_{nn}^*} - \sum_{k=n+1}^r \frac{Y_{nk}^* \mathbf{V}_k^{*(p)}}{Y_{nn}^*} + \frac{P_n}{\mathbf{V}_n^{(p)} Y_{nn}^*} \quad (24)$$

can be calculated immediately and substituted into (23):

$$\mathbf{V}_n^{*(p+1)} = (\mathbf{V}_n^{*(p+1)})^\# + \frac{jQ_n}{\mathbf{V}_n^{(p)} Y_{nn}^*} \quad (25)$$

but, for a generator node, the magnitude $|V_n|$ is known, so considering the magnitudes in (25):

$$|V_n|^2 = \left[\Re \left\{ (\mathbf{V}_n^{*(p+1)})^\# + \frac{jQ_n}{\mathbf{V}_n^{(p)} Y_{nn}^*} \right\}^2 + \Im \left\{ (\mathbf{V}_n^{*(p+1)})^\# + \frac{jQ_n}{\mathbf{V}_n^{(p)} Y_{nn}^*} \right\}^2 \right] \quad (26)$$

which can be solved for Q_n (by iteration if necessary). The calculated value of Q_n is substituted back into (25) and the new estimate of generator node voltage is found.

When compared to the Newton-Raphson Method, the Gauss-Seidel Method involves simple calculations, but is slow to converge. Therefore, it is common practice to accelerate the iterative process, by adding to the newly-calculated value of each variable an extra term proportional to the difference between the new and previous values. For example:

$$\mathbf{V}_n^{*(p+1)} \Big|_{\text{accelerated}} = \mathbf{V}_n^{*(p+1)} + \alpha \cdot \{ \mathbf{V}_n^{*(p+1)} - \mathbf{V}_n^{*(p)} \} \quad (27)$$

where α is an 'acceleration factor', which has a typical value of 0.6.

4.2. Application of the Gauss-Seidel Method to the Specific Problem

For the particular 3-node problem, introduced in Section 2, Equation (23) can be used directly to calculate the voltage at the load node (node 3) every iterative cycle.

To deal with the generator node (node 2), multiply (25), with $n=2$, by $\mathbf{V}_2^{(p)} Y_{22}^*$:

$$\mathbf{V}_2^{(p)} Y_{22}^* \mathbf{V}_2^{*(p+1)} = \mathbf{V}_2^{(p)} Y_{22}^* (\mathbf{V}_2^{*(p+1)})^\# + jQ_2 \quad (28)$$

and expressing the phasor voltages in polar form:

$$|V_2| \angle \delta_2^p |Y_{22}| \angle -\theta_{22} |V_2| \angle -\delta_2^{p+1} = |V_2| \angle \delta_2^p |Y_{22}| \angle -\theta_{22} |V_2^\#| \angle \delta_2^\# + jQ_2 \quad (29)$$

the term jQ_2 is imaginary and can be eliminated by equating real parts in (29):

$$|V_2| |Y_{22}| |V_2| \cos\{\delta_2^p - \delta_2^{p+1} - \theta_{22}\} = |V_2| |Y_{22}| |V_2^\#| \cos\{\delta_2^p + \delta_2^\# - \theta_{22}\} \quad (30)$$

$$\therefore |V_2| \cos\{\delta_2^p - \delta_2^{p+1} - \theta_{22}\} = |V_2^\#| \cos\{\delta_2^p + \delta_2^\# - \theta_{22}\} \quad (31)$$

$$\therefore \delta_2^{p+1} = \delta_2^p - \theta_{22} - \arccos\left\{\frac{|V_2^\#|}{|V_2|} \cos\{\delta_2^p + \delta_2^\# - \theta_{22}\}\right\} \quad (32)$$

4.3. Sample Results for the Gauss-Seidel Method

The Gauss-Seidel Method, defined by (23,24,25,27,32), for solving the load flow problem outlined in Section 2 is implemented in Sheet 4 of the Workbook. Sample results for the default input data and using an acceleration factor $\alpha = 0.6$ are shown in Table 2.

Comparing the results from Table 2 with those from Table 1, the slower convergence of the Gauss-Seidel Method is evident: even after 10 iteration cycles the values of phase angle are stabilised only to a single decimal place. However the computational effort involved in each cycle is much reduced in the Gauss-Seidel Method. Nevertheless both iterative methods require some complicated mathematical operations: inversion of the large Jacobian matrix in the Newton-Raphson Method and intermediate calculation of reactive power input at generator nodes in the Gauss-Seidel Method.

5. Stochastic Search Techniques

5.1. General Principles

Recent developments in load flow analysis have moved attention away from the iterative methods and towards so-called stochastic search methods. Two such methods – Genetic Algorithms and Simulated Annealing – are described here and are implemented in the Workbook. Both approaches use a series of trial solutions to the problem and develop better solutions in the light of experience gained from these trials. The computational effort for each trial is kept as low as possible, so a very large number of trials can be conducted.

For the example problem being considered throughout this paper, there are three variables: the voltage magnitude $|V_3|$ and the phase angles δ_2, δ_3 . In any one trial some appropriate values are chosen for these variables. The choice may be an entirely random selection across the entire possible range of values (termed the ‘search space’) or the choice may be informed by previous experience. Once these trial values are chosen, the phasor voltages at all three nodes are defined, because all of the other voltage magnitudes and phase angles are fixed. Therefore the

currents injected at each node can be evaluated directly using (2) and the corresponding complex power input is calculated using (5).

The success of the trial needs to be judged by some quantitative criterion. The trial values of node voltage lead to values of input power and reactive power (P , Q) that do not exactly match the pre-defined values (P_s , Q_s). The extent of the mis-match can be quantified conveniently, for this particular problem, with the error function:

$$E = \sqrt{(P_2 - P_{2s})^2 + (P_3 - P_{3s})^2 + (Q_3 - Q_{3s})^2} \quad (33)$$

The stochastic search techniques use the error function to inform the selection of new potential solutions for the subsequent round of trials. It is this selection process which is defined by the particular search technique.

5.2. Genetic Algorithms

Genetic Algorithms imitate the process of evolution, where the fittest individuals are likely to survive in a competing environment. A genetic algorithm starts with a random population of potential individuals, or chromosomes, each representing one possible solution to a problem. The chromosomes are simply a collection of genes, each gene being one of the solution variables. The chromosomes are then evolved through successive generations. During each generation, all the chromosomes are evaluated, according to a defined fitness criterion, and the best chromosomes are selected to mate and generate offspring. The least fit chromosomes of each population are then replaced by the offspring so that the population size remains constant. After several generations, the algorithm converges to the best chromosome which represents an optimal solution to the problem.

A further refinement of the evolution process, again mirroring nature, is that any chromosome in any generation has a finite probability of suffering mutation, in which some of the genes are randomly perturbed. It is this process which ensures that the genetic algorithm does not converge to a local minimum when searching for a global problem solution.

When applied to the load flow problem, the genes are the nodal voltage magnitude and phase angle values and each chromosome contains a complete set of the genes needed to define uniquely a trial solution. The fitness of each chromosome is evaluated using the error criterion (33), which is used as the basis of selection for the

chromosomes in the next generation. For the example problem, a genetic algorithm solution is implemented in Sheet 5 of the Excel Workbook. Results from the common example problem during the early stages of the algorithm are shown in Table 3. An initial population of 10 chromosomes forms the first generation. Chromosome 1 is placed at the centre of the search space, chromosomes 2-9 are located at the extremes of the search space and chromosome 10 has random values. The values of the genes, voltage magnitude $|V_3|$ and the phase angles δ_2, δ_3 , for each chromosome are shown in rows 1-3 of the Table. Each chromosome, in association with the pre-defined values of $|V_1|, |V_2|$ and δ_1 , defines a set of trial nodal voltages, which are used to calculate the input currents (rows 4-6) and the power / reactive power inputs (rows 7-9). The set values for the power inputs are ($P_{2s} = 1.0$; $P_{3s} = -0.5$; $Q_{3s} = -0.2$), and the error function defined in (33) is evaluated for each chromosome at row 10 with a low value of error function indicating a closer match between the calculated and set power values. The chromosomes are then ranked (row 11) by error. Rows 1-11 represent one complete generation of the genetic algorithm's evolutionary process.

Chromosomes for the second generation are derived from the previous generation's chromosomes according to their ranking. Row 12 indicates how each chromosome in the second generation has been formed. The first two chromosomes are copies of the two highest-ranked chromosomes of the previous generation. Chromosomes 3-5 are obtained by breeding between chromosomes 1 and 2: in each case one gene in chromosome 1 is replaced by the corresponding gene from chromosome 2. Each gene in chromosome 6 takes the average value of the genes in chromosomes 1 and 2. Mutation takes place in chromosomes 7-9, with one gene in turn from chromosome 1 being replaced by a random value. Finally, chromosome 10 of the second generation has genes which are entirely randomly-selected. So the new generation has 10 chromosomes, most of which are related to the best chromosomes from the previous generation.

The cycle of chromosome evaluation and breeding continues through many generations. Evaluation of the error function for each chromosome requires relatively simple calculations, so the genetic algorithm can continue through many generations until the error has fallen to acceptable levels. A typical variation of error with generation number is shown in Fig. 3. The genetic algorithm quickly reduces the error, but a very large number of generations is needed to bring the error close to zero. In the Excel Workbook, accelerated convergence is obtained after 100 generations, by redefining the search space so that it is centred on the best available solution at that stage and is reduced in size. The effect of this redefinition is apparent in Fig. 3, where the error reduction receives fresh impetus after the 100th generation.

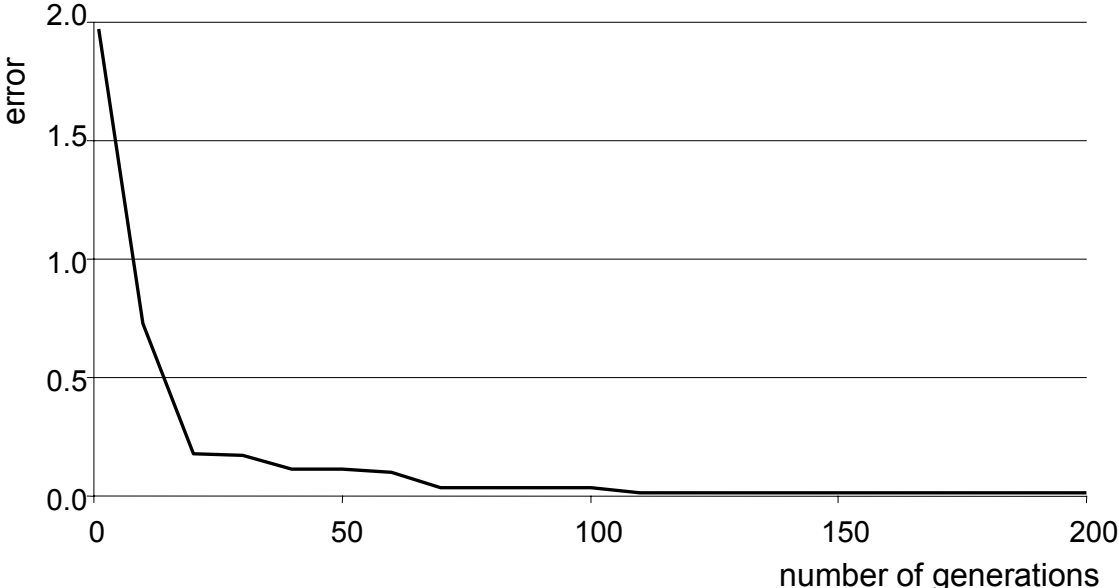


Fig. 3 Typical error associated with the best chromosome as a function of the number of generations

5.3 Simulated Annealing

Simulated annealing is a global search technique in which a randomly-generated potential solution, Y, to a problem is compared to an existing solution, X. The probability of Y being accepted for investigation depends on the proximity of Y to X and the extent to which the solution has been developed, as represented by a 'temperature' parameter, T, which reduces throughout the annealing process. Both potential solutions are investigated and Y is chosen to replace X as the existing

solution according to a probability function, which again depends on the temperature T.

To apply this concept to load flow studies in general, it is assumed that the solutions X and, Y, represent information about possible nodal voltage values. In the particular illustrative example defined in Section 2, there are three unknown voltage values, so the solutions can be written:

$$X = [\delta_{2X}, |V_3|_X, \delta_{3X}] \quad Y = [\delta_{2Y}, |V_3|_Y, \delta_{3Y}] \quad (34)$$

where these solutions must lie within the pre-defined search space. Table 4 presents sample results from the simulated annealing technique, which is implemented in Sheet 6 of the Workbook. The solution commences on row 1 with an initial best value, which is placed at the centre of the search space and its error E_X evaluated using (33).

The range indicates the extent of the search space and is defined as:

$$range = \frac{\sqrt{(\delta_{2MAX} - \delta_{2MIN})^2 + (|V_3|_{MAX} - |V_3|_{MIN})^2 + (\delta_{3MAX} - \delta_{3MIN})^2}}{2} \quad (35)$$

A new set of voltage values, Y, are selected at random. The displacement is the distance between X and Y in the search space:

$$displacement = \sqrt{(\delta_{2Y} - \delta_{2X})^2 + (|V_3|_Y - |V_3|_X)^2 + (\delta_{3Y} - \delta_{3X})^2} \quad (36)$$

An acceptance probability, P_A , is then calculated:

$$P_A = \exp \left[- \left(\frac{displacement}{range} \right) * \left(\frac{T_s}{T} \right) \right] \quad (37)$$

where T is the instantaneous temperature and T_s is the initial temperature. P_A is compared to a random value r_1 in the range [0 – 1]. If $r_1 > P_A$, the potential solution Y is rejected and the process is repeated. If $r_1 < P_A$ then Y is accepted for evaluation. Note that for small values of displacement and high values of temperature, the acceptance probability is close to unity, so potential solutions are more likely to be accepted for evaluation if they are close to the existing solution or if the temperature is high. For example, in the first row of Table 4, the displacement between X and Y is small (0.10) and the temperature is equal to the initial temperature. Therefore the

calculated acceptance probability is high (0.83). The random number $r_1 = 0.15$ and so Y is accepted for evaluation.

When Y is accepted for evaluation the corresponding nodal voltages are used to calculate the input currents {from (2)} and the complex power inputs {from (5)}. Hence the error function E_Y can be found from (33). This error is compared to the error E_X obtained for the solution X in the swap probability function, P_S :

$$P_S = \frac{1}{1 + \exp\left\{\left(\frac{E_Y - E_X}{E_X}\right) * \left(\frac{T_S}{T}\right)\right\}} \quad (38)$$

P_S is compared to a random value r_2 in the range [0 – 1]. If $r_2 > P_S$, then the original solution X is retained, and, if $r_2 < P_S$, the new solution Y, is accepted, and X is replaced by Y. Substitution of Y for X is most likely to occur if the error E_Y is small, in which case the swap probability is high. However the temperature T also influences the likelihood of swapping. In row 1 of Table 4 the values of E_Y and E_X are almost equal and the swap probability is 0.51. However the random number r_2 is 0.49, so a swap of Y for X does occur. Therefore in the second row of Table 4 the ‘current best values’ are the Y values from row 1.

Looking more generally at Table 4, in row 2 the next set of random values Y are accepted for evaluation, but produce an error E_Y (3.19) which is substantially larger than E_X (1.93), resulting in a low swap probability of 0.33. The value of r_2 generated is 0.68, so a swap does not occur. Note, however, that there is a finite probability of a ‘worse’ value being substituted. The probability of such an event happening diminishes as the temperature reduces. Whenever a random value is too far displaced from the current best value, it may not be accepted for evaluation, as happens in row 3. The current and power calculations do not need to be made in these circumstances.

Many alternatives for the decrement function of the temperature, T, are available. In this work, the temperature is inversely proportional to the number of potential solutions investigated. Convergence is assisted by re-defining the search space and resetting the temperature to its initial value after every 1000 values have been considered for acceptance. Fig. 4 shows how the error from the best values tends to reduce with the number of potential solutions investigated. However there are instances where the error increases. Even though the error produced by a new set of

trial values may be larger than the best value error, there is a finite probability that a swap will occur, resulting in an increased best value error. In Fig. 4 the increases in error tend to occur when the temperature is high.

Comparing the two search methods, genetic algorithms and simulated annealing, the number of calculations required to produce an acceptable error is similar: for the default power system parameters the genetic algorithm operates over 200 generations, with 8 new chromosomes to be evaluated in each generation, giving a total of 1600 evaluations. Simulated annealing requires in the order of 700 potential solutions to be investigated. If a random search was conducted across the entire search space then to obtain comparable resolution in the solution would require the evaluation of approximately 300 voltage magnitude values and 200 values of each phase angle, giving a total of $300 \times 200 \times 200 = 1.2 \times 10^7$ evaluations.

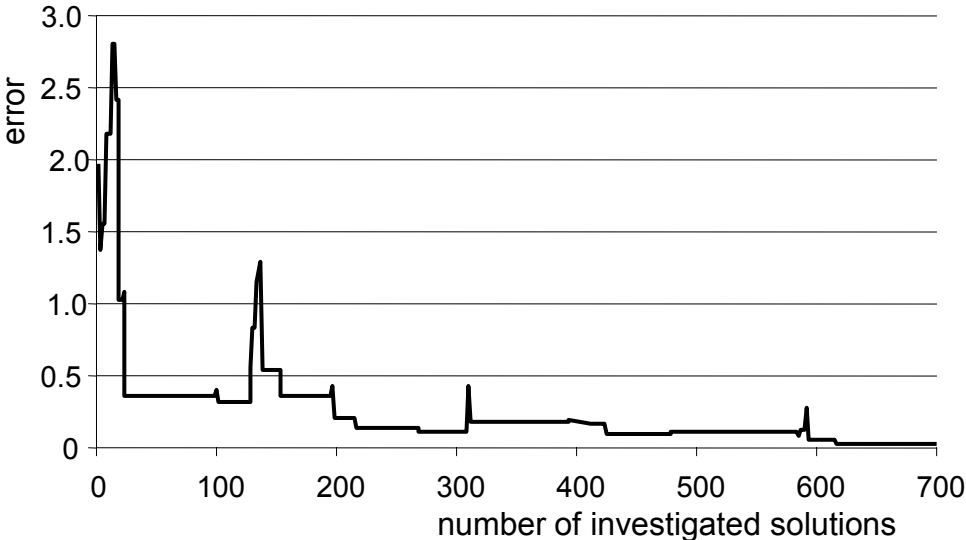


Fig. 4 Variation of 'best value' error E_x with the number of solutions accepted for evaluation

6. Summary of Results and System Calculations

Sheet 7 of the Excel Workbook summarises the results of the nodal voltage calculations from the four methods. The error from the final best value from the two search methods is also presented, so the user is alerted to any failure to arrive at a solution with the appropriate accuracy. If necessary, the calculation cycle can be repeated by pressing the (F9) key.

The ultimate purpose of load flow studies is to calculate the flow of power and reactive power through the system. Therefore, included in the Sheet is the calculation of the power and reactive power flow through each of the three lines and a calculation of the total power and reactive power consumed by the transmission system. (Note that the default system parameters define a lossless system, so the power consumed is zero.)

7. Conclusions

Four separate methods of power system load flow analysis have been implemented in an MS-Excel Workbook, which allows students to appreciate the operation of the numerical methods without needing advanced computational skills. The Workbook can be used for two different aspects of engineering education: i) as an introduction to load flow studies for power systems students, in which case the focus of attention is the system data (Sheet 2) and the load flow results on Sheet 7, without being concerned with the calculation methods; ii) as an example problem for courses introducing numerical methods of problem solving, using both iterative methods (Sheets 3 and 4) and stochastic search methods (sheets 5 and 6).

Iteration	δ_2°	$ V_3 $	δ_3°
0	10.000	0.900	-20.000
1	-2.813	1.242	4.773
2	-0.346	1.093	-3.822
3	-0.226	1.052	-5.210
4	-0.224	1.050	-5.278
5	-0.224	1.050	-5.278

Table 1: Results from the Newton-Raphson method

Iteration	δ_2°	$ V_3 $	δ_3°
0	10.000	0.900	-20.000
1	-6.156	1.027	-9.985
2	-3.752	1.046	-7.805
3	-1.251	1.048	-5.935
4	-0.770	1.050	-5.651
5	-0.552	1.049	-5.490
6	-0.333	1.050	-5.333
7	-0.354	1.050	-5.379
8	-0.250	1.050	-5.325
9	-0.229	1.049	-5.271
10	-0.177	1.049	-5.216

Table 2: Results from the Gauss-Seidel Method

	1 st Generation Chromosome No.	1	2	3	4	5	6	7	8	9	10
1	δ_2 (radians)	0.000	0.175	0.175	0.175	0.175	-0.175	-0.175	-0.175	-0.175	0.107
2	$ V_3 $	1.000	1.100	1.100	0.900	0.900	1.100	1.100	0.900	0.900	0.937
3	δ_3 (radians)	0.000	0.175	-0.175	0.175	-0.175	0.175	-0.175	0.175	-0.175	0.106
4	l_1	0.0+ 0.4j	-1.72 +0.75j	0.19 +0.75j	-1.55 -0.24j	0.017 -0.24j	-0.19 +0.75j	1.72 +0.75j	-0.017 -0.24j	1.55 -0.24j	-0.97 +0.034j
5	l_2	0.0 +0.4j	-1.72 +0.75j	0.19 +0.75j	-1.55 -0.24j	0.017 -0.24j	-0.19 +0.75j	1.72 +0.75j	-0.017 -0.24j	1.55 -0.24j	0.92 +0.37j
6	l_3	0.0 +1.0j	0.95 -0.42j	-4.78 -0.42j	0.43 +2.54j	-4.25 +2.54j	4.78 -0.42j	-0.96 -0.42j	4.25 +2.538j	-0.43 +2.54j	1.08 +1.57j
7	P_2	0.000	0.764	4.902	0.764	4.150	-4.902	-0.764	-4.150	-0.764	-1.035
8	P_3	0.000	0.955	-5.093	0.781	-4.167	5.093	-0.955	4.167	-0.781	1.115
9	Q_3	-1.000	0.633	1.363	-2.182	-1.585	1.363	0.633	-1.585	-2.182	-1.462
10	E	1.972	2.603	5.530	3.031	4.354	8.986	2.026	7.782	2.748	3.546
11	ranking	1	3	8	5	7	10	2	9	4	6
12		best	2nd best	breed from best	breed from best	breed from best	breed from best	mutate best	mutate best	mutate best	random
13	2 nd Generation Chromosome No.	1	2	3	4	5	6	7	8	9	10
14	δ_2 (radians)	0.000	-0.175	0.000	0.000	-0.175	-0.087	0.000	0.000	0.008	0.099
15	$ V_3 $	1.000	1.100	1.000	1.100	1.000	1.050	1.000	0.977	1.000	0.972
16	δ_3 (radians)	0.000	-0.175	-0.175	0.000	0.000	-0.087	-0.034	0.000	0.000	0.075

Table 3: Operation of the Genetic Algorithm during the first generation

Count	Current best values				New random values				Acceptance			Calculate Currents		Calculate Power			Swap				
	δ_{2x} (radians)	V_{3x}	δ_{3x} (radians)	E_x	T_s/T	δ_{2Y} (radians)	V_{3Y}	δ_{3Y} (radians)	displacement	P_A	random, r_1	accept ?	I_2	I_3	P_2	P_3	Q_3	E_Y	P_S	random, r_2	swap ?
1	0.00	1.00	0.00	1.97	1.0	0.02	1.10	0.02	0.10	0.83	0.15	Y	0.11 -0.43j	0.06 -0.45j	0.11	0.06	0.50	1.93	0.51	0.49	Y
2	0.02	1.10	0.02	1.93	1.1	-0.03	0.92	0.02	0.18	0.69	0.49	Y	-0.61 -2.18j	0.60 +2.17j	-0.61	0.59	-1.99	3.19	0.33	0.68	N
3	0.02	1.10	0.02	1.93	1.2	-0.16	0.98	-0.02	0.22	0.61	0.70	N									
4	0.02	1.10	0.02	1.93	1.2	-0.03	1.00	-0.14	0.19	0.66	0.62	Y	0.93 -1.45j	-1.76 +1.08j	1.07	-1.90	-0.83	0.75	0.68	0.41	Y
5	-0.03	1.00	-0.14	0.75	1.3	0.13	0.92	0.12	0.32	0.46	0.14	Y	0.83 -2.16j	0.30 +2.23j	0.60	0.52	-2.00	2.74	0.03	0.13	N
6	-0.03	1.00	-0.14	0.75	1.4	-0.03	1.01	-0.08	0.06	0.86	0.19	Y	0.35- 1.35j	-0.90 +0.94	0.43	-0.98	-0.86	1.02	0.38	0.66	N
7	-0.03	1.00	-0.14	0.75	1.5	-0.07	0.96	0.12	0.27	0.47	0.50	N									
8	-0.03	1.00	-0.14	0.75	1.5	-0.08	1.06	0.11	0.26	0.48	0.50	N									
9	-0.03	1.00	-0.14	0.75	1.5	-0.02	0.92	0.06	0.21	0.55	0.46	Y	-0.82 -2.21j	1.00 +2.22j	-0.86	1.04	-1.99	3.62	0.00	0.35	N

Table 4: Typical initial results from Simulated Annealing