A FAST DECOUPLED SPARSITY ORIENTED ALGORITHM FOR THE SOLUTION OF THE LOAD FLOW PROBLEM IN ELECTRIC POWER SYSTEMS

by

Prof. S. Karunaratne, BSc (Eng), MSc (Eng), CEng, MIEE, MIE (SL)
Dr. H. Sriyananda, BSc (Eng), MSc, MSc, PhD, CEng, MIEE
Dr. J. R. Lucas, BSc (Eng), MSc, PhD, Mem IEE
Mr. A. S. Induruwa, BSc (Eng)
Mr. B. S. P. Perera, BSc (Eng)

Abstract

The Newton-Raphson method is one of the fastest converging algorithms available for the solution of nonlinear equations. The Load Flow problem is first expressed in a form suitable for the implementation of this method. This is then modified by incorporating the decoupling property, between real power and nodal voltage and, between reactive power and transmission angle, inherent in practical high voltage power systems. Further simplifications are affected by justifiable assumptions on the orders of magnitudes of the node voltages and transmission angles, which lead to an invariant Jacobian matrix. Economy in both storage requirements and computation time required for the resulting Fast Decoupled (Newton-Raphson) Algorithm is achieved by exploiting the sparse nature of the Jacobian matrix by using a list-processing type organisation of the computer memory for the recording of and operations of this matrix. Triangular factorization is considered for the solution of the Newton-Raphson equations, as being more appropriate under these circumstances than conventional inversion methods. The effects of reordering on the conservation of sparsity are also discussed. The Fast Decoupled Sparsity Oriented Algorithm developed is suitable for the study of medium-sized systems using small computers. A case study of a 41 bus, 47 line system using a 16 K machine is presented.

\[ \theta_{km} = \theta_k - \theta_m \] — angle difference across nodes k and m.
\[ \nabla V, \nabla \theta \] — voltage magnitude and angle corrections.
\[ \text{PQ} \] — type of busbar where active and reactive power are specified.
\[ \text{PV} \] — type of busbar where active power and voltage magnitude are specified.
\[ \text{SLACK} \] — reference busbar, where voltage magnitude and angle are specified.
\[ m \theta_k \] — topological operator which signifies that bus m is connected to bus k, including the case m—k.
\[ \text{p} \] — signifies a vector or matrix.
\[ [ \text{p} ] \] — iteration count.

1.0 Introduction

The attention of Power System Engineers has been drawn towards the problem of Load Flow, for a long period of time. However, until the advent of the digital computer, load flow problems of significant magnitude have never been solved. The application of the computer has opened almost a new dimension in the field of load flow problems and their solution. Problems of very large magnitude have been solved using the computer.

In the early stages, interest has been in the development of algorithms for the solution of the problem and less attention has been given to the time required for solution. In the next stage, the attention has been mainly on the speed of solution, by developing better algorithms. Finally, the effort has been mainly on the economic use of the storage capacity of the computer.

This paper describes a method for the solution of the load flow problem which economises on the storage capacity of the computer and solution time. The Newton-Raphson method is used as it gives better convergence and the 'decoupling' is used in order to save on computer storage. Further, a modified form of the N-R algorithm is used where Jacobian matrices are invariant from one
iteration to the other in order to cut down further on solution time and the property of the sparsity of the Jacobian matrix is exploited to further economise on the storage and time.

The result is a 'Fast Decoupled Newton-Raphson with Sparsity' technique which is suitable to solve large-scale load flow problems using small computers.

2.0 Fast Decoupled Load Flow Algorithm

2.1 Newton-Raphson (N-R) Method

The Newton-Raphson method involves the solution of a set of non-linear equations \( F(X) = 0 \), where \( F \) is a vector of functions, \( f_1, f_2, \ldots, f_n \) in the variables \( x_1, x_2, \ldots, x_n \). At each iteration of the solution, the non-linear problem is approximated by the linear matrix equation

\[
F(X^{p-1}) = -J(X^{p-1})\Delta X^p
\]

where \( \Delta X \) is the correction vector, and \( p \) is the iteration count. The square matrix \( J \) is the Jacobian of \( F(X) \) and its \((k,m)\)th element is defined as

\[
\frac{\partial f_k}{\partial x_m}
\]

Equation (2.1) is solved for \( \Delta X^p \), and \( X \) is updated using

\[
X^p = X^{p-1} + \Delta X^p
\]

The method converges very rapidly if the initial estimates are good and \( F(X) \) is well behaved.

2.2 Application of N-R method to the load flow problem

Using the standard notation, the complex power injected at node \( k \) is

\[
P_k + jQ_k = V_k \left( \cos \theta_k + j \sin \theta_k \right)
\sum_{m \in k} Y_{km} V_m^* F_m^*
\]

which equation may be written in polar form as:

\[
P_k + jQ_k = V_k \left( G_{km} \cos \theta_{km} + j B_{km} \sin \theta_{km} \right)
\sum_{m \in k} \left( G_{km} \cos \theta_{km} + j B_{km} \sin \theta_{km} \right)
\]

separating the real and the imaginary parts

\[
P_k = V_k \sum_{m \in k} V_m \left( G_{km} \cos \theta_{km} + B_{km} \sin \theta_{km} \right)
\]

\[
Q_k = V_k \sum_{m \in k} V_m \left( G_{km} \sin \theta_{km} + B_{km} \cos \theta_{km} \right)
\]

From the above, it is possible to define the power mismatch equations for the three types of busbars as follows:

For a PQ busbar

\[
\Delta P_k = P_k^{\text{p}} - V_k \sum_{m \in k} V_m \left( G_{km} \cos \theta_{km} + B_{km} \sin \theta_{km} \right)
\]

\[
\Delta Q_k = Q_k^{\text{p}} - V_k \sum_{m \in k} V_m \left( G_{km} \sin \theta_{km} - B_{km} \cos \theta_{km} \right)
\]

For a PV busbar

Only equation (2.7) is available as the reactive power is not specified.

For SLACK busbar

No expression for \( \Delta P_k \) and \( \Delta Q_k \) are available as neither the real power nor the reactive power is specified.

Considering the power mismatch equations, equation (2.1) becomes

\[
\begin{bmatrix}
\Delta P^{p-1} \\
\Delta Q^{p-1}
\end{bmatrix}
= \begin{bmatrix}
H^{p-1} & N^{p-1} \\
J^{p-1} & L^{p-1}
\end{bmatrix}
\begin{bmatrix}
\Delta \theta^p \\
\Delta V^p
\end{bmatrix}
\]

where the elements of the submatrices are

\[
\begin{align*}
H_{km} &= -\frac{\partial P_k}{\partial \theta_m}, & N_{km} &= -\frac{\partial P_k}{\partial V_m} \\
J_{km} &= -\frac{\partial Q_k}{\partial \theta_m}, & L_{km} &= -\frac{\partial Q_k}{\partial V_m}
\end{align*}
\]

Expressions for these elements are derived in Appendix A1. As the elements of the Jacobian matrix are dependent on \( V \) and \( \theta \), these need to be evaluated at each iteration. In addition, the Jacobian matrix has also to be inverted at each iteration.

2.3 The Fast Decoupled Load Flow Method

The disadvantages of the N-R method, such as the need for repeated evaluation and inversion of the Jacobian matrix, may be eliminated if certain physical properties of the system under study are incorporated. For an electrical power system, the important properties are the loose physical interaction between real and reactive power flows in a power system resulting from the high \( X/R \) ratio. High \( X/R \) ratios of transmission lines is a characteristic feature of a high voltage transmission system.

Consideration of the above properties simplifies the algorithm by permitting the \( P-V \) and \( Q-\theta \) coupling submatrices, namely \( H \) and \( J \) in equation (2.9) to be neglected.

This then gives rise to the two separate reduced equations

\[
\begin{bmatrix}
\Delta P^{p-1} \\
\Delta Q^{p-1}
\end{bmatrix}
= \begin{bmatrix}
H^{p-1} & N^{p-1} \\
L^{p-1} & V^{p-1}
\end{bmatrix}
\begin{bmatrix}
\Delta \theta^p \\
\Delta V^p
\end{bmatrix}
\]

\[
\begin{bmatrix}
\Delta P^{p-1} \\
\Delta Q^{p-1}
\end{bmatrix}
= \begin{bmatrix}
H^{p-1} & N^{p-1} \\
L^{p-1} & V^{p-1}
\end{bmatrix}
\begin{bmatrix}
\Delta \theta^p \\
\Delta V^p
\end{bmatrix}
\]
Figure 1 - Logic Flow Chart for the Fast Decoupled Load Flow Method
These two equations could be solved alternately (decoupled N-R method) re-evaluating (H) and (L) at each iteration.

However, as in an actual system, the following assumptions are almost always valid, further physically justifiable simplifications may be made.

Since \( \theta_{km} \) is generally very small,

\[
\cos \theta_{km} \approx 1, \quad \mathrm{and} \quad G_{km} \sin \theta_{km} \ll B_{km}
\]

the equations (A1.4) and (A1.8), in the appendix, simplify to

\[
H_{km} = I_{km} = V_k (-B_{km}) V_m
\]

and neglecting \( Q_k \), equations (A1.9) and (A1.10) simplify to

\[
H_{kk} = I_{kk} = V_k (-B_{kk}) V_k
\]

With the above simplifications, the general terms of the matrix equations (2.10) and (2.11) may be written as

\[
\Delta P_k = \sum_{m \in \mathcal{E}} V_k B_{km} V_m \Delta \theta_m \quad (2.12)
\]

and

\[
\Delta Q_k = \sum_{m \in \mathcal{E}} V_k B_{km} V_m' \Delta V_m \quad (2.13)
\]

In the above equations, \( V_k \) is independent of the summation variable and may be taken over to the left hand side of the equation. Also since \( P-V \) decoupling is assumed, \( V_m \) in equation (2.12) may be put equal to \( \gamma \) per unit.

The equations (2.12) and (2.13) may thus be written as

\[
\frac{\Delta P_k}{V_k} = -B_{km} \Delta \theta_m
\]

\[
\frac{\Delta Q_k}{V_k} = -B_{mk} \Delta V_m
\]

or written (with the iteration count also included) in matrix form these become

\[
\begin{bmatrix}
\frac{\Delta P^{n-1}}{V^{n-1}} \\
\frac{\Delta Q^{n-1}}{V^{n-1}}
\end{bmatrix} = (B) \begin{bmatrix}
\Delta \theta^n \\
\Delta V^n
\end{bmatrix}
\]

(2.14)

and

\[
\begin{bmatrix}
\frac{\Delta P^{n-1}}{V^{n-1}} \\
\frac{\Delta Q^{n-1}}{V^{n-1}}
\end{bmatrix} = (B) \begin{bmatrix}
\Delta \theta^n \\
\Delta V^n
\end{bmatrix}
\]

(2.15)

The matrix \( B \) is real, sparse and, as it contains only network susceptance values, the values of its elements are constants. Therefore, this matrix need be formed only once, that is, at the beginning of computation. This set of equations is then the Fast Decoupled Load Flow equations.

2.4 Iteration Scheme

The logic flow chart of the fast decoupled load flow is given in figure 1. The best iteration scheme is to solve equations (2.14) and (2.15) alternately. Each iteration cycle comprises one solution for \( \theta \) and one solution for \( V \) and is called the \( (\theta, V) \) scheme. This avoids any possibility of overconvergence of \( \Delta P \) and \( \Delta Q \) respectively towards exact solution these equations, each have geometric convergence rates and is not as fast as the \( N-R \)'s quadratic rate. However, this fact is compensated by the faster iteration speed of the fast decoupled method.

2.5 Convergence Criterion

Separate convergence testing is used for equations (2.14) and (2.15) with the criteria

\[
\max \frac{\Delta P}{P} \leq c_p, \quad \max \frac{\Delta Q}{Q} \leq c_q
\]

where \( \max \frac{\Delta P}{P} \) and \( \max \frac{\Delta Q}{Q} \) are the maximum real power mismatch and the maximum reactive power mismatch respectively, and \( c_p \) and \( c_q \) are the chosen convergence tolerances for real and reactive power respectively. The values of \( c_p \) and \( c_q \) can be read in according to individual requirements.

3.0 Sparsity Programming in Load Flow Studies

In addition to the decoupling of \( P \) and \( Q \) and \( \theta \), there is another property of electrical power systems that may be exploited for the development of an efficient computer program for the solution of the load flow problem. This is the fact that the bus admittance matrix is a very sparse one. This is a property of all practical systems since any particular node is connected to only a small number of other nodes.

There are two advantages to be gained by storing and operating on only the non-zero elements of the \( B \) matrix. These are

1. Saving in storage
2. Saving in time

3.1 Storage Economy

Consider an electrical power system with \( n \) nodes and \( l \) lines. If the bus admittance matrix is to be stored in the usual matrix form, it would require \( n^2 \) locations. However, as there are only \( l \) lines, the total number of non-zero values of this matrix would be \( (2l+n) \). Also, since \( y_{ij} = y_{ji} \), the total number of elements that need to be stored is only \( (l+n) \). In general, this is very much less than \( n^2 \), and considerable savings in storage requirements may be effected by storing only the non-zero elements. A number of methods have been suggested for optimising the memory for this purpose, and one that has been widely used is described below.

1. Storage of diagonal elements: As the diagonal elements of \( B \) are always non-zero, they are stored as a vector of size \( n \).

2. Storage of off-diagonal elements: These are stored in an array of blocks, each block consisting of three words of storage (Figure 2).
The first element of each block contains the index of the column in which the element appears, the second contains the value of the element and the third is a pointer to the location where the next block corresponding to the next non-zero element of the same row is located. If it is the last non-zero element of the row, the pointer is set to zero. Thus, a total storage of three times the number of elements is required.

An extra array of dimension \(n\) is used as an indexing array to point to the location of the first non-zero off-diagonal element of each row. Thus, it is possible to store the elements in any location, and not necessarily arranged in any particular order.

3.2 Time economy

It can be achieved by avoiding the extra complexity in accessing data, as operations are performed only on non-zero elements.

3.3 Triangular factorization and Gaussian elimination

The solution of the load flow problem was shown to reduce to the solution of the set of matrix equations

\[
\begin{pmatrix}
\Delta P \\
\Delta Q
\end{pmatrix} = - \begin{bmatrix} B \end{bmatrix} \cdot \begin{bmatrix} \Delta \theta \\
\Delta V
\end{bmatrix}
\]

and

\[
\begin{pmatrix}
\Delta V
\end{pmatrix} = - \begin{bmatrix} B \end{bmatrix} \cdot \begin{bmatrix} \Delta V
\end{pmatrix}
\]

If these equations are solved by matrix inversion, it would nullify the advantages of sparsity programming as inversion would, in general, fill up an originally sparse matrix. Instead, Gaussian elimination by triangular factorization is adopted, as this method conserves sparsity.

In the triangular factorization algorithm, a slightly different procedure from the standard Gaussian elimination method is used, especially in the matter of storing intermediate results. This is because it is desired to obtain an operator-type entity which can be used with any vector to obtain a solution of an equation of the form shown above.

Figure 3 shows the contents of the elements after factorization.

It is possible to use the factorized matrix as an operator on any vector by forward and backward substitution. With symmetrical matrices (as in the case with the bus admittance matrices), only one half of the matrix need be stored, and elimination may be carried out either by rows or by columns. It has been found that columnwise elimination is, in general, more efficient.

3.4 Optimal ordering for conservation of sparsity

It can be shown that the number of new off-diagonal elements introduced during the factorization process depends on the ordering of the equations. It also depends on the topology of the particular system, but in general, the following schemes in increasing order of merit conserve sparsity:

1. reorder to ensure that the number of off-diagonal elements in each row are in ascending order.
2. select first row as above, then insert new elements that would be introduced during triangular factorization with this row, and then select the row with the next largest number of off-diagonal elements as the second. Repeat for third, fourth row etc.
3. at each stage, select row that would introduce the least number of new non-zero elements as the next row.

4.0 Conclusion

The application of the Newton-Raphson method of solution to the Power System load flow study has been described in this paper. It has been suggested how, due to the high \(X/R\) ratios of transmission lines of high voltage systems, decoupling of \(P\) with \(V\), and \(Q\) with \(\theta\), may be effected for simpler evaluation and lesser storage. It is also shown how the repeated evaluation and inversion of the Jacobian matrix necessary in the normal Newton-Raphson has been avoided by exploiting this property. Although the method then requires slightly more iter-
A FAST DECOUPLED SPARSITY ORIENTED ALGORITHM

For a given accuracy, it is still more efficient in computer time due to the lesser number of calculations required at each step.

Further improvements are effected by utilizing the sparsity of the susceptance matrix caused by the small number of branches connected to each node. The algorithm utilizes the storage of, and operation on, only the non-zero elements. It is thus economical in computation and provides a large reduction in computer storage, enabling small computers to be used for the evaluation of comparatively large size electrical power systems.

The method has been illustrated using the Sri Lanka Power System on an IBM 1130 Computer (core memory 16 K words).

1.0 Acknowledgements

The authors are grateful to their students who have helped in the development and running of the computer programs associated with this paper...

6.0 References


APPENDIX A

Derivation of the elements of the Jacobian matrix

Consider the complex power injected at a node $k$

$$ P_k + j Q_k = E_k \Gamma_k \star $$  \hspace{1cm} (A1.1)

The current injected at the node $k$ is given by

$$ I_k = \sum_{m \in K} Y_{km} E_m $$

where $m \in K$ is the topological operator.

Using polar notation, the voltage at node $m$ is given by

$$ E_m = V_m (\cos \theta_m + j \sin \theta_m) $$

and its conjugate

$$ E_m \star = V_m (\cos \theta_m - j \sin \theta_m) $$

Let $Z_{km} = R_{km} + j X_{km}$ be the impedance of branch $k,m$ then

$$ Y_{km} = \frac{R_{km}^2}{R_{km}^2 + X_{km}^2} - j \frac{X_{km}}{R_{km}^2 + X_{km}^2} $$

Hence, from equation (A1.1),

$$ P_k + j Q_k = E_k \sum_{m \in K} Y_{km} E_m $$

$$ = V_k (\cos \theta_k + j \sin \theta_k) \sum_{m \in K} \left( G_{km} - j B_{km} \right) V_m (\cos \theta_m - j \sin \theta_m) $$

$$ = V_k \sum_{m \in K} V_m (\cos \theta_m + j \sin \theta_m) $$

$$ \left( G_{km} - j B_{km} \right) $$

where $\theta_{km} = \theta_k - \theta_m$

Thus,

$$ P_k + j Q_k = V_k \sum_{m \in K} V_m \left[ \left( G_{km} \cos \theta_{km} + B_{km} \sin \theta_{km} \right) \right. $$

$$ \left. + j \left( G_{km} \sin \theta_{km} - B_{km} \cos \theta_{km} \right) \right] $$

Separating the real and the imaginary parts,

$$ P_k = V_k \sum_{m \in K} V_m \left( G_{km} \cos \theta_{km} + B_{km} \sin \theta_{km} \right) $$

and

$$ Q_k = V_k \sum_{m \in K} V_m \left( G_{km} \sin \theta_{km} - B_{km} \cos \theta_{km} \right) $$

Let the complex power specified at node $k$ be

$$ \Delta P_k = P_k^{sp} - P_k $$

then the real power mismatch is given by

$$ \Delta P_k = P_k^{sp} - P_k $$

i.e., $\Delta P_k = P_k^{sp} - V_k \sum_{m \in K} V_m \left( G_{km} \cos \theta_{km} + B_{km} \sin \theta_{km} \right)$$

$$ + V_k \sum_{m \in K} V_m \left( G_{km} \cos \theta_{km} + B_{km} \sin \theta_{km} \right) $$

$$ \frac{1}{m \neq k} $$
\[ P_k^{sp} - V_k^2 G_{kk} + V_k \sum_{m \neq k} V_m \left( G_{km} \cos \theta_{km} + B_{km} \sin \theta_{km} \right) \]

substituting from equation (A1.2), the above expression reduces to:

\[ H_{kk} = -V_k^2 B_{kk} - Q_k \]

Similarly,

\[ L_{kk} = -V_k \frac{\partial}{\partial V_k} \left( Q_k^{sp} + V_k^2 B_{kk} - V_k \sum_{m \neq k} V_m \left( G_{km} \sin \theta_{km} - B_{km} \cos \theta_{km} \right) \right) \]

where the elements of the Jacobian matrix are defined by

\[ H_{km} = \frac{\partial H_k}{\partial \theta_m}, \quad L_{km} = -\frac{\partial L_k}{\partial V_m}, \quad N_{km} = -V_m \frac{\partial P_k}{\partial V_m}, \quad I_{km} = -V_m \frac{\partial Q_k}{\partial V_m}. \]

Since there is only one line connecting node k to node m, H_{km} is a single element, and for all k not connected to m, G_{km} and B_{km} are both equal to zero. Hence the summation signs in equations (A1.3) and (A1.4) can be omitted in the evaluation of H_{km}

Thus for k ≠ m

\[ H_{km} = \frac{\partial}{\partial \theta_m} \left( P_k^{sp} - V_k^2 V_m \left( G_{km} \cos \theta_{km} + B_{km} \sin \theta_{km} \right) \right) = V_k V_m \left( G_{km} \sin \theta_{km} - B_{km} \cos \theta_{km} \right) \]

\[ N_{km} = -V_m \frac{\partial P_k}{\partial V_m} \left( G_{km} \cos \theta_{km} + B_{km} \sin \theta_{km} \right) \]

Similarly,

\[ J_{km} = -V_k V_m \left( G_{km} \cos \theta_{km} + B_{km} \sin \theta_{km} \right) \]

\[ L_{km} = -V_k V_m \left( G_{km} \cos \theta_{km} - B_{km} \cos \theta_{km} \right) \]

Now for k = m

\[ H_k = \frac{\partial}{\partial \theta_k} \left( P_k^{sp} - V_k^2 G_{kk} + V_k \sum_{m \neq k} V_m \left( G_{km} \cos \theta_{km} + B_{km} \sin \theta_{km} \right) \right) \]

\[ N_k = V_k \sum_{m \neq k} V_m \left( G_{km} \sin \theta_{km} - B_{km} \cos \theta_{km} \right) \]

\[ I_k = -V_k \sum_{m \neq k} V_m \left( G_{km} \sin \theta_{km} - B_{km} \cos \theta_{km} \right) \]

\[ L_k = -V_k \sum_{m \neq k} V_m \left( G_{km} \cos \theta_{km} + B_{km} \cos \theta_{km} \right) \]

The data input to the computer is shown enclosed in the above table.
### Case Study: Proposed 41 Bus Sri Lanka Power System for 1985

- System base = 40 MVA,
- System voltage = 132 kV,
- Number of lines = 47,
- Number of busbars = 41,
- Maximum acceptable MW/MVAR mismatch = 0.01 p.u.

<table>
<thead>
<tr>
<th>Line</th>
<th>Sending bus</th>
<th>Receiving bus</th>
<th>Shunt admittance</th>
<th>Series impedance</th>
<th>Real Power</th>
<th>Reactive Power</th>
<th>Line Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LAX132</td>
<td>POL132</td>
<td>0+j0.0394</td>
<td>0.0008+j0.0018</td>
<td>2.805</td>
<td>0.450</td>
<td>0.0078</td>
</tr>
<tr>
<td>2</td>
<td>POL132</td>
<td>LAX132</td>
<td>0+j0.1320</td>
<td>0.0126+j0.0288</td>
<td>-0.136</td>
<td>-0.342</td>
<td>0.0022</td>
</tr>
<tr>
<td>3</td>
<td>UKU132</td>
<td>POL132</td>
<td>0+j0.0394</td>
<td>0.0033+j0.0075</td>
<td>1.138</td>
<td>0.204</td>
<td>0.0041</td>
</tr>
<tr>
<td>4</td>
<td>ORUWAL</td>
<td>KOL132</td>
<td>0+j0.0060</td>
<td>0.0066+j0.0130</td>
<td>-0.797</td>
<td>-0.323</td>
<td>0.0028</td>
</tr>
<tr>
<td>5</td>
<td>LAXST1</td>
<td>LAX 66</td>
<td>0+j0.0000</td>
<td>0.0066+j0.0130</td>
<td>0.600</td>
<td>0.381</td>
<td>0.0028</td>
</tr>
<tr>
<td>6</td>
<td>LAX 66</td>
<td>LAXST1</td>
<td>0+j0.0060</td>
<td>0.0066+j0.0130</td>
<td>-0.797</td>
<td>-0.323</td>
<td>0.0028</td>
</tr>
<tr>
<td>7</td>
<td>AVISA W</td>
<td>AVISA W</td>
<td>0+j0.0213</td>
<td>0.0393+j0.0732</td>
<td>0.252</td>
<td>0.076</td>
<td>0.0023</td>
</tr>
<tr>
<td>8</td>
<td>AVISA W</td>
<td>LAX 66</td>
<td>0+j0.0000</td>
<td>0.0066+j0.0130</td>
<td>-0.797</td>
<td>-0.323</td>
<td>0.0028</td>
</tr>
<tr>
<td>9</td>
<td>BALANG</td>
<td>AGALAW</td>
<td>0+j0.1500</td>
<td>0.0066+j0.0130</td>
<td>0.600</td>
<td>0.381</td>
<td>0.0028</td>
</tr>
<tr>
<td>10</td>
<td>AGALAW</td>
<td>BALANG</td>
<td>0+j0.1500</td>
<td>0.0066+j0.0130</td>
<td>0.600</td>
<td>0.381</td>
<td>0.0028</td>
</tr>
<tr>
<td>11</td>
<td>DENIYA</td>
<td>DENTIA</td>
<td>0+j0.1500</td>
<td>0.0066+j0.0130</td>
<td>0.600</td>
<td>0.381</td>
<td>0.0028</td>
</tr>
<tr>
<td>12</td>
<td>LAX132</td>
<td>LAX 66</td>
<td>0+j0.0000</td>
<td>0.0066+j0.0130</td>
<td>0.600</td>
<td>0.381</td>
<td>0.0028</td>
</tr>
<tr>
<td>13</td>
<td>RATMAL</td>
<td>KOL132</td>
<td>0+j0.0102</td>
<td>0.0035+j0.0093</td>
<td>0.418</td>
<td>0.347</td>
<td>0.0010</td>
</tr>
<tr>
<td>14</td>
<td>AGALAW</td>
<td>RATMAL</td>
<td>0+j0.0102</td>
<td>0.0035+j0.0093</td>
<td>0.418</td>
<td>0.347</td>
<td>0.0010</td>
</tr>
<tr>
<td>15</td>
<td>BALANG</td>
<td>BDU132</td>
<td>0+j0.0710</td>
<td>0.0035+j0.0093</td>
<td>0.418</td>
<td>0.347</td>
<td>0.0010</td>
</tr>
<tr>
<td>16</td>
<td>BDU 66</td>
<td>NUELW</td>
<td>0+j0.0224</td>
<td>0.0390+j0.0740</td>
<td>0.154</td>
<td>0.010</td>
<td>0.0033</td>
</tr>
<tr>
<td>17</td>
<td>NUELW</td>
<td>BALANG</td>
<td>0+j0.0224</td>
<td>0.0390+j0.0740</td>
<td>0.154</td>
<td>0.010</td>
<td>0.0033</td>
</tr>
<tr>
<td>18</td>
<td>NUELW</td>
<td>NORTON</td>
<td>0+j0.0224</td>
<td>0.0390+j0.0740</td>
<td>0.154</td>
<td>0.010</td>
<td>0.0033</td>
</tr>
<tr>
<td>19</td>
<td>LAX 66</td>
<td>NORTON</td>
<td>0+j0.0224</td>
<td>0.0390+j0.0740</td>
<td>0.154</td>
<td>0.010</td>
<td>0.0033</td>
</tr>
<tr>
<td>20</td>
<td>NORTON</td>
<td>LAX 66</td>
<td>0+j0.0224</td>
<td>0.0390+j0.0740</td>
<td>0.154</td>
<td>0.010</td>
<td>0.0033</td>
</tr>
<tr>
<td>21</td>
<td>LAX 66</td>
<td>ER YAG</td>
<td>0+j0.0224</td>
<td>0.0390+j0.0740</td>
<td>0.154</td>
<td>0.010</td>
<td>0.0033</td>
</tr>
<tr>
<td>22</td>
<td>ER YAG</td>
<td>LAX 66</td>
<td>0+j0.0224</td>
<td>0.0390+j0.0740</td>
<td>0.154</td>
<td>0.010</td>
<td>0.0033</td>
</tr>
<tr>
<td>23</td>
<td>ER YAG</td>
<td>NORTON</td>
<td>0+j0.0224</td>
<td>0.0390+j0.0740</td>
<td>0.154</td>
<td>0.010</td>
<td>0.0033</td>
</tr>
<tr>
<td>24</td>
<td>KOI 132</td>
<td>SAPUGS</td>
<td>0+j0.0023</td>
<td>0.0047+j0.0093</td>
<td>0.940</td>
<td>0.168</td>
<td>0.0028</td>
</tr>
<tr>
<td>25</td>
<td>SAPUGS</td>
<td>KOL 132</td>
<td>0+j0.0023</td>
<td>0.0047+j0.0093</td>
<td>0.940</td>
<td>0.168</td>
<td>0.0028</td>
</tr>
<tr>
<td>26</td>
<td>BOL WAT</td>
<td>SAPUGS</td>
<td>0+j0.0023</td>
<td>0.0047+j0.0093</td>
<td>0.940</td>
<td>0.168</td>
<td>0.0028</td>
</tr>
<tr>
<td>27</td>
<td>BOL WAT</td>
<td>PUT LAM</td>
<td>0+j0.0023</td>
<td>0.0047+j0.0093</td>
<td>0.940</td>
<td>0.168</td>
<td>0.0028</td>
</tr>
<tr>
<td>28</td>
<td>RAJ ANG</td>
<td>PUL AM</td>
<td>0+j0.0023</td>
<td>0.0047+j0.0093</td>
<td>0.940</td>
<td>0.168</td>
<td>0.0028</td>
</tr>
<tr>
<td>29</td>
<td>ANURAD</td>
<td>RAJ ANG</td>
<td>0+j0.0023</td>
<td>0.0047+j0.0093</td>
<td>0.940</td>
<td>0.168</td>
<td>0.0028</td>
</tr>
<tr>
<td>30</td>
<td>RAJ ANG</td>
<td>ANURAD</td>
<td>0+j0.0023</td>
<td>0.0047+j0.0093</td>
<td>0.940</td>
<td>0.168</td>
<td>0.0028</td>
</tr>
<tr>
<td>31</td>
<td>ANURAD</td>
<td>TRIN CO</td>
<td>0+j0.0023</td>
<td>0.0047+j0.0093</td>
<td>0.940</td>
<td>0.168</td>
<td>0.0028</td>
</tr>
<tr>
<td>32</td>
<td>TRIN CO</td>
<td>ANURAD</td>
<td>0+j0.0023</td>
<td>0.0047+j0.0093</td>
<td>0.940</td>
<td>0.168</td>
<td>0.0028</td>
</tr>
<tr>
<td>Line</td>
<td>SB</td>
<td>RB</td>
<td>Y_{sh}</td>
<td>Z_{sc}</td>
<td>P</td>
<td>Q</td>
<td>P_L</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>--------</td>
<td>--------</td>
<td>-----------</td>
<td>-------</td>
<td>-------</td>
<td>--------</td>
</tr>
<tr>
<td>26</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.1600</td>
<td>0.0531+j0.1220</td>
<td>0.303</td>
<td>0.007</td>
<td>0.0047</td>
</tr>
<tr>
<td>27</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.1600</td>
<td>0.0531+j0.1220</td>
<td>0.303</td>
<td>0.007</td>
<td>0.0047</td>
</tr>
<tr>
<td>28</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.0822</td>
<td>0.0274+j0.0626</td>
<td>0.252</td>
<td>0.073</td>
<td>0.0016</td>
</tr>
<tr>
<td>29</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.0822</td>
<td>0.0274+j0.0626</td>
<td>0.252</td>
<td>0.073</td>
<td>0.0016</td>
</tr>
<tr>
<td>30</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.114</td>
<td>0.0211+j0.0375</td>
<td>0.080</td>
<td>0.025</td>
<td>0.0001</td>
</tr>
<tr>
<td>31</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.0822</td>
<td>0.0274+j0.0626</td>
<td>0.252</td>
<td>0.073</td>
<td>0.0016</td>
</tr>
<tr>
<td>32</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.114</td>
<td>0.0211+j0.0375</td>
<td>0.080</td>
<td>0.025</td>
<td>0.0001</td>
</tr>
<tr>
<td>33</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.2370</td>
<td>0.0196+j0.0450</td>
<td>0.310</td>
<td>0.269</td>
<td>0.0021</td>
</tr>
<tr>
<td>34</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.1300</td>
<td>0.0150+j0.0344</td>
<td>1.452</td>
<td>0.060</td>
<td>0.0277</td>
</tr>
<tr>
<td>35</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.1300</td>
<td>0.0150+j0.0344</td>
<td>1.452</td>
<td>0.060</td>
<td>0.0277</td>
</tr>
<tr>
<td>36</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.1300</td>
<td>0.0150+j0.0344</td>
<td>1.452</td>
<td>0.060</td>
<td>0.0277</td>
</tr>
<tr>
<td>37</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.1300</td>
<td>0.0150+j0.0344</td>
<td>1.452</td>
<td>0.060</td>
<td>0.0277</td>
</tr>
<tr>
<td>38</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.1300</td>
<td>0.0150+j0.0344</td>
<td>1.452</td>
<td>0.060</td>
<td>0.0277</td>
</tr>
<tr>
<td>39</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.1300</td>
<td>0.0150+j0.0344</td>
<td>1.452</td>
<td>0.060</td>
<td>0.0277</td>
</tr>
<tr>
<td>40</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.1300</td>
<td>0.0150+j0.0344</td>
<td>1.452</td>
<td>0.060</td>
<td>0.0277</td>
</tr>
<tr>
<td>41</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.1300</td>
<td>0.0150+j0.0344</td>
<td>1.452</td>
<td>0.060</td>
<td>0.0277</td>
</tr>
<tr>
<td>42</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.1300</td>
<td>0.0150+j0.0344</td>
<td>1.452</td>
<td>0.060</td>
<td>0.0277</td>
</tr>
<tr>
<td>43</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.1300</td>
<td>0.0150+j0.0344</td>
<td>1.452</td>
<td>0.060</td>
<td>0.0277</td>
</tr>
<tr>
<td>44</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.1300</td>
<td>0.0150+j0.0344</td>
<td>1.452</td>
<td>0.060</td>
<td>0.0277</td>
</tr>
<tr>
<td>45</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.1300</td>
<td>0.0150+j0.0344</td>
<td>1.452</td>
<td>0.060</td>
<td>0.0277</td>
</tr>
<tr>
<td>46</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.1300</td>
<td>0.0150+j0.0344</td>
<td>1.452</td>
<td>0.060</td>
<td>0.0277</td>
</tr>
<tr>
<td>47</td>
<td>KILO</td>
<td>KILO</td>
<td>0+j0.1300</td>
<td>0.0150+j0.0344</td>
<td>1.452</td>
<td>0.060</td>
<td>0.0277</td>
</tr>
</tbody>
</table>

Total Line Loss — 0.630 p.u.

*The Fast Decoupled Load Flow with Sparsity* technique converged, with specified accuracy in 6 iterations.