Chapter-III
CHAPTER - III

A CONSTANT MATRIX METHOD FOR LOAD FLOW
IN RECTANGULAR COORDINATES

3.1 Introduction

Load flow analysis leading to the solution of the steady state operating conditions of an electric power transmission system, is the starting step for the solutions of a number of power system problems. Results of the load flow equations are required for the system planning, the operational planning and control, for large systems state estimation and outage security assessment and also for the more complicated stability and optimisation computations. Load flow analysis, thus is the bread & butter of the modern power systems.

The increasing availability of high speed digital computers has brought about a dramatic change in the techniques used to solve the power system load flow problem. Prior to 1930, power flow computations were made manually. But the subsequent introduction of analogue based network analyzers enabled more rapid solutions to be obtained. The development of electronic digital computers during the 1950’s provided an alternative means of solution and in 1956, Ward and Hale [183] described the first practical computer oriented solution technique for the load flow problem employing nodal iterative method. It had the advantage of simplicity of analysis and small computer memory requirements. Earlier methods provided varying degree of automation of the load flow computations but did not present an integrated method which was independent of operator intervention once the computation was initiated and they were based on the Gauss-Seidel algorithm using the admittance form of the network equations. However, with the rapid growth in the size of power system networks during the 1960’s,
the inherent convergence difficulties of the method abounded. The search for alternative
techniques resulted in the development of Newton-Raphson (NR) algorithm in a form suitable
for the solution of the simultaneous quadratic equations describing the power system models.
The NR method is shown to be superior, in most cases, to the Gauss-Seidel approach
provided that good estimates of the initial nodal voltages are available. The major disadvantage
of this method is the requirement of increased computer memory due to the Jacobian matrix
needed to direct the iterations. However, the improved convergence and the availability of
computers with large core storage has resulted in the acceptance of the Newton-Raphson
method as a preferred technique.

Numerous other techniques have since been described and modifications and extensions
to existing methods are being published regularly [47-49], [70], [75], [87], [129], [136], [137], [182].
The types of iterative techniques and the variations on them are numerous, each having its
own advantage. Although some iterative techniques seem to be preferred much over others,
their effectiveness is sometimes a function of the characteristics of a particular power system
being analyzed and thus, it is almost impossible to provide an unbiased test for their
effectiveness.

The algorithms and programs to solve the power flow problem have steadily improved
over the last three decades in terms of speed and reliability. It is usually possible to classify
the load flow solution techniques into the following groups based upon the frame of reference
adopted.

- Bus Impedance Matrix Methods
- Bus Admittance Matrix Methods and
- Hybrid Methods
3.1.1 Bus Impedance Matrix Methods

The first practical automatic digital solution method [31] employs Y-matrix iterative methods to solve load flow equations since they require minimal computer storage. Although they perform satisfactorily on many problems, they converge slowly, and too often, do not converge at all. This deficiency led to the development of Z-matrix Gauss-Seidel iterative methods [19-20], [57] which converge more rapidly but sacrifice some of the advantages of Y-matrix iterative methods; mainly, the storage requirement and the speed when applied to large systems. In the Z-matrix method, linear equations in terms of the bus impedance parameters is solved. A technique of great general value is to represent a portion of each bus load as a fixed shunt admittance, which in effect reduces the value of non-linear bus power constraints [19]. Brameller [15] considers the nodal equations for all buses excluding the slack bus. A pair of quadratic equations is solved to attain convergence. It is also found that the Z-matrix methods are usually not too sensitive to the choice of slack bus. Ward and Hale [183] solve linearized versions of the quadratic equations at each iteration. This method is similar to the above method [19] and a direct relationship between the new nodal voltages and the previous voltage profile was established. The solution of the load flow problem was initiated by assuming voltages for all buses except the slack bus for which the voltage is fixed.

The increase in the size of problems has led to the development of methods employing diakoptical by Happ [68] and block-iteration techniques by Bosaroge et. al. [14] for load flow. Happ [68] proved that the network tearing method of Kron [94] can be used to solve the load flow problem in conjunction with the Newton-Raphson method with tearing performed either before or after the linearisation of the system equations. In the first case the number of iterations is identical to the untorn case, because network tearing is an exact method for linear systems. In the second case, where the non-linear equation are torn, Happ found that
extra iterations are introduced in the Newton-Raphson process. Sasson [148] has shown that the diakoptical method can be applied to the load flow in either nodal admittance or nodal impedance formulations and confirms Happ’s finding on the introduction of extra iterations.

3.1.2 Bus Admittance Matrix Methods

Among the other methods developed in 1950’s is the one proposed by Glimn and Stagg [62] wherein a Gauss-Seidel approach is used in which the non-linear bus constraints are incorporated by substituting for the unknown currents. During that time, the Newton-Raphson method was developed and was shown to have very powerful convergence properties, but was found to be computationally uncompetitive. In the mid 1960’s Tinney [173] and others developed a very efficient sparsity program using the ordered elimination technique. The major achievement of this method is the fastness of the solution and reduced storage requirements due to which the Newton’s method is being widely used as a general purpose load flow solution technique. With increasing problem size, on-line application, and system optimisation, newer methods have been developed and have found wider acceptance in the power industry. The Newton-Raphson method attempts to combine the fast convergence property comparable to the impedance method and low computer memory requirements associated with the sparsity of admittance matrix method. This is accomplished by combining the sparsity programming and a Jacobian or rate of change matrix.

Aggarwal et. al. [1] developed a load flow solution technique based on Newton-Richardson method. Tripathy et. al. [174] used K.M.Brown’s method for load flow solution of ill conditioned systems. They used a variation of Newton’s method incorporating a Gaussian elimination in such a way that the most recent information is always used at each step of the algorithm. Iwamato et. al. [83] developed a load flow model based on NR approach and without using any mathematical approximations. This load flow solution never diverges and the existence of solution can be easily judged. Sutanto [171] discussed the convergence properties of NR load flow when a three winding transformer is present. Araposthatis et. al.
[8] studied the quantitative properties of the power flow equations for transmission network in terms of global and local aspects as well as stability. Global aspects include estimates of the number of solutions and topological properties of the stable region. Local aspects are examined through bifurcation of load flow equations. Aschmoneit et. al. [9] proposed a method which calculates in advance the system reaction to change in switching transformer tap position and load. Irving et. al. [79] proposed a NR based partitioned matrix approach to the Jacobian equation. This method works well for low voltage and ill conditioned system. Roy [142] used a cartesian coordinate formulation, exact, constant, real, sparse and diagonally dominant coefficient matrix of the size same as that of FDLF model for his algorithm. El-Hawary et. al. [49] studied the effect of high capacitance with high reactive power on the Newton’s method.

Regardless of the quality of the load flow program, there are system models that do not converge. This condition arises due to variety of reasons. The following factors normally characterize an ill-conditioned system:-

- network configurations (longitudinal network and large R/X ratio)
- operating conditions (heavy loads)
- high ratio of PQ/PV buses

Most of the decoupled load flow methods are derived from the network method by neglecting the coupling sub-Jacobians. For a system having high R/X ratio for all branches, the convergence pattern of the decoupled load flow method may be dictated by the coupling sub-Jacobian rather than the diagonal sub-Jacobian [69]. Amongst the decoupled versions, the most popular one is the fast decoupled load flow (FDLF) method of Stott and Alsac [169], because it is computationally the fastest and, for most of the power systems, is reliable. Wu [188] reported that the FDLF, however, does have convergence problems on systems with branches that have large resistance to reactance ratios. Haque [70] proposed a novel way of decoupling the Newton load flow method without ignoring most of the elements of
the coupling sub-Jacobian. The effects of the coupling sub-Jacobian are intelligently incorporated in the mismatch vectors and diagonal sub-Jacobians. It may be noticed that in the FDLF method the B' and B" matrices are fixed as long as there is no change in system topology and/or the status of regulated buses. However, both matrices have to be updated when the system topology changes due to line outages etc. and due to changes like Q-limit violation, Generator outage, addition of a new generator etc., the status of regulated buses affect the B" matrix only.

Rajicic and Bose [129] presented a modification of FDLF for networks with high R/X ratios. In this method some coefficients are calculated to update the existing B' and B" matrices to improve the convergence problem for system with large R/X ratios. But this method was found to be slightly slower than the FDLF method for systems with normal ratios. In early 1986, Chang and Brandwajn [28] introduced two methods for updating triangular factorized matrices such as B' and B". These methods effectively resolve the problem created by the need for updating B'. The method is known as Partial refactorisation method (PRI). The method is efficient when the status of a few buses change, but when there is a topology change due to a large number of buses (large outages) this method becomes inefficient, because the efficiency of PRI method is inversely proportional to the number of bus status changes. To overcome the difficulty of updating of the B" matrix, a new method know as the Nodal Iterative Model was proposed. This method is efficient because it does not use Jacobian like matrices. Instead, it uses relations between voltages, powers and admittances. However, this method is extremely slow in convergence.

A general purpose version of FDLF was developed in 1989 by Van Amerogen [178], in which it was suggested that the resistance be ignored in B" matrix instead of in B' matrix. Van Amerogen [178] studied the convergence behaviour of four different decoupled load flow algorithms. He designated the algorithms as BB, BX, XB and XX. Where each of the two letters indicate the coefficient or Jacobian matrix of the (P-δ) and (Q-V) equations respectively. The letter X indicates that the branch susceptances are obtained by neglecting
the branch resistance, and letter B indicates that the branch susceptances are obtained by considering both the branch resistances and susceptances. Van Amerogen [178] found that both the BX and XB versions have almost the same convergence behaviour for most well behaved systems having relatively low R/X ratio branches. However BX versions exhibit poorer convergence behaviour than XB versions for ill conditioned systems. This finding is further confirmed by Hubbi [75] and he also found that for heavily loaded systems the XB version is more reliable than the BX version. A decoupled power flow solution method was proposed by El-Arini [47] which can solve both well and ill conditioned power systems, and requires less storage and computation time as compared to either the NR method or the FDLF method. The power flow equations are decoupled into P and Q model without any approximations and second order vectors are added to the real and reactive power with some multipliers. But as this method stores full matrices, it requires larger computer memory. A similar conclusion is arrived at by Nanda et. al. [120], who state that the decoupled load flow method will have poor convergence behaviour when the branch resistances are considered (or neglected) in both the P-δ and Q-V problems. To obtain good convergence behaviour in the decoupled load flow methods, the branch resistances have to be considered either in the P-δ or Q-V problem, but not in both. Several other techniques for decoupling the load flow equations for networks having high R/X ratio branches are also reported in the literature [38], [67], [118], [124].

Several important work on second order load flow methods have been attempted in the past which consider more exact models for reliable and faster solutions. Consideration of the second order terms in the Taylor's series expansion of the nonlinear power flow equations in polar form was first studied by Sachdev and Medicherla [146]. This version also suffers from a large memory requirement and greater computational time as the second order terms are computed repeatedly in each iteration; also, corrective vectors are computed twice in each iteration. Nanda et. al. [119] developed a second order FDLF method in polar coordinates and used a totally different approach from those employed in the existing second
order methods in polar coordinates. Their work eliminates the need for storing and computing repeatedly the second order terms by prudently injecting the elements of the Hessian matrix into the Jacobian. This method sometimes fails for certain ill-conditioned systems. Iwamato and Tamura [83] were the first to suggest a second order load flow method in rectangular coordinates. A fast load flow retaining non-linearity using rectangular coordinates was proposed which can solve well conditioned systems but fails for some ill-conditioned systems. Later Roy [143], Rao et. al. [136-137] and Nanda et. al. [118] suggested improved versions of the same. The second order rectangular coordinate versions are unable to achieve proper decoupling and hence require larger memory as compared to the polar version. They also take more time but give accurate results. The second order methods take more input/output time as compared to the FDLF method because the equations for the formation of the Jacobian matrix are not as straightforward as the FDLF model. So the total time and memory increases with the system size.

3.1.3 Hybrid Methods

A hybrid Newton/Minimisation method was proposed by Sasson et. al. [147], which attempts to accelerate Newton's method. After one iteration, a straight line through the first and second iteration points in the multidimensional coordinate space should pass reasonably close to the solution point if the process is converging. This method guarantees non divergence, but the interpretation of a located minimum \((f_0=0)\) is difficult. The method proposed by Dusonchet et. al. [38] uses Z-matrix method for PQ buses and polar power-mismatch Newton versions to solve for PV buses. A block successive iteration is performed between Newton and Z-matrix method. But the drawback of this method is the increased number of iterations required as compared to Newton's method. A hybrid model of FDLF developed by Behnamguilani [12] is found to be faster than normal FDLF model and resolves the matrix updating problem. This is a decoupled nodal iterative method and combines the active network of the FDLF with the reactive network of the nodal iteration model. It is slow in convergence.
for some systems, because of the active network model developed.

Galloway et. al. [60] developed a digital step-by-step transient-response calculation, which is continued until the system reaches its steady state, and this is the solution to the system load flow problem. Though the method is highly reliable, it is not competitive with the other methods. Deckman et. al. [36] developed a similar modification using parallel/series branch compensation. These compensation methods, however, give mixed results in that the improvement in convergence is not consistent. This method is used to reduce artificially the R/X ratio of a branch; it increases the system size and possesses a much slower convergence pattern. Wallach [181] proposed a method in which the load flow problem is converted into the minimization of an unconstrained scalar objective function which is usually equal to the sum of the squares of the currents or power mismatches. The non-linear programming field provides many minimum seeking processes from which to choose; but this process was not found suitable for the basic load flow method. Pan [122] used optimal control theory based algorithm for load flow solution and shows that mismatch of the load flow equations follows a monotonically decreasing trajectory. Keyhani et. al. [88] proposed a knowledge based power flow model.

From the results of the literature survey, it is concluded that the bus impedance matrix method of load flow have not become popular because of heavy storage required. The FDLF method does not work well for large R/X ratio systems, and second order load flow versions takes more time and memory. For the hybrid method, more computations per iteration are required.

Constant matrix methods of load flow are also available in the literature. A constant matrix method in rectangular coordinates is proposed by Singh et. al. [158]. The matrix used in this algorithm is the same as the matrix of the polar FDLF method of Stott and Alsac [169] and therefore it is likely to have similar convergence problems for ill-conditioned systems. Prasad et. al. [127] also proposed a constant matrix method in rectangular coordinates in which the constant matrix is different from the constant matrix of Singh et. al. [158] and
showed that the convergence is faster and more reliable in the sense that solution converges even for ill-conditioned systems. The line resistances are also included in the constant matrix proposed by Prasad et. al. [127].

In this Chapter a constant matrix method for load flow in rectangular coordinates is discussed. A relationship is also established between the inverse of the constant matrix and the system bus impedance matrix to reduce the overall solution time. This method is generally fast and performs well even on ill-conditioned systems as the information of the line resistance is also retained in the constant matrix.

The proposed method is explained in Section 3.2. In Section 3.3 the relationships between the constant matrix and the $Z_{bus}$ is presented. Details of the system studies carried out and the results are given in Section 3.4. The salient features of the proposed algorithm are discussed in Sections 3.5. Finally, the concluding remarks are given in Section 3.6.

3.2 Proposed Constant Matrix Method

In this Section, starting with the usual load flow equations in rectangular coordinates a constant matrix model is derived and the performance of this method is compared with the polar and rectangular versions of FDLF and polar version of constant Jacobian method. The last bus is selected as the reference bus.

The complex power equations of a power system network is given by

$$P_i + jQ_i = V_i \sum (YV_i)'$$

or

$$P_i + jQ_i = (e_i + jf_i)[(Ge - Bf)_i - j(Be + Gf)_i]$$

where $(YV_i)'$ is the conjugate of the $i^{th}$ component of the vector $[YV]$.

The complex power can be broken up into real and imaginary parts as

$$P_i = e_i(Ge - Bf)_i + f_i(Be + Gf)_i$$

and

63
\[ Q_i = f_i(Ge - Bf_i) - e_i(Be + Gf_i) \]  \hspace{1cm} (3.4)

Also,
\[ |V_i| = \sqrt{e_i^2 + f_i^2} \]  \hspace{1cm} (3.5)

Since, the values of \( e \) and \( f \) at the slack bus, \( P \) and \( Q \) at all the PQ buses and \( P \) and \( V \) at all the PV buses are specified, we now have to solve for the real and imaginary parts of the voltages at all buses except the slack bus.

\( P_i \) is calculated for \( i=1,2...N-1 \) and \( Q_i \) is calculated for \( i=1,2,3......N-NPV-1 \).

Let \( \Delta P_i, \Delta Q_i, \Delta e_i \) and \( \Delta f_i \) denote the incremental changes in the respective quantities.

Then
\[ \Delta P_i = \Delta e_i(Ge - Bf_i) + e_i(G \Delta e - B \Delta f)_i + \]
\[ \Delta f_i(Be + Gf_i) + f_i(B \Delta e + G \Delta f)_i \]  \hspace{1cm} (3.6)

\[ \Delta Q_i = -e_i(B \Delta e + G \Delta f)_i - \Delta e_i(Be + Gf_i) + \]
\[ \Delta f_i(Ge + Bf_i) + f_i(G \Delta e - B \Delta f)_i \]  \hspace{1cm} (3.7)

\[ 0 = 2e_i \Delta e_i + 2f_i \Delta f_i \]  \hspace{1cm} (3.8)

Rewriting (3.6) and (3.7) as
\[ \frac{\Delta P_i}{e_i} = (G \Delta e - B \Delta f)_i + \frac{\Delta e_i}{e_i}(Ge - Bf_i) + \]
\[ \frac{\Delta f_i}{e_i}(Be + Gf_i) + \frac{f_i}{e_i}(B \Delta e + G \Delta f) \]  \hspace{1cm} (3.9)

\( i = 1, 2, 3.....N-1; \)

\[ \frac{\Delta Q_i}{e_i} = -(B \Delta e + G \Delta f)_i - \frac{\Delta e_i}{e_i}(Be + Gf_i) + \]
\[ \frac{\Delta f_i}{e_i}(Ge - Bf_i) + \frac{f_i}{e_i}(G \Delta e - B \Delta f) \]  \hspace{1cm} (3.10)

\( i = 1, 2, 3..... N-NPV-1; \)

The load flow equations can be written in terms of Jacobian matrix as
\[
\begin{bmatrix}
\Delta P \\
\Delta Q
\end{bmatrix} =
\begin{bmatrix}
J_1 & J_2 \\
J_3 & J_4
\end{bmatrix}
\begin{bmatrix}
\Delta f \\
\Delta e
\end{bmatrix}
\] (3.11)

where

\[
\Delta P = [\Delta P_1, \ldots, \Delta P_{n-1}]^T
\]
\[
\Delta Q = [\Delta Q_1, \ldots, \Delta Q_{n-npv-1}]^T
\]
\[
\Delta f = [\Delta f_1, \ldots, \Delta f_{n-npv-1}]^T
\]
\[
\Delta e = [\Delta e_1, \ldots, \Delta e_{n-1}]^T
\]

Jacobians \( J_1, J_2, J_3 \) and \( J_4 \) are functions of \( e \) and \( f \), and have to be evaluated in each iteration as \( e \) and \( f \) get modified in each iteration. To make the Jacobian a constant matrix, the following assumptions are made.

1. \( e_i \) is approximately equal to 1.0 and \( f_i \) is approximately equal to zero for all \( i=1,2,\ldots,N \).

2(a) \( B \) is a symmetric and diagonally dominant matrix such that

\[
\sum_{j=1}^{N} B_{ij} = 0
\]

Thus \( Be \) (in eq. 3.9) is approximately equal to zero, and \( Bf \) (in eq. 3.9) is approximately equal to zero from assumption 1.

2(b) \( G \) is also a symmetric and diagonally dominant matrix such that

\[
\sum_{j=1}^{N} G_{ij} = 0
\]

Hence \( Ge \) is approximately zero and \( Gf \) is also approximately zero from assumption 1.

With the above assumptions, equations (3.9) and (3.10) can be simplified to

\[
\frac{\Delta P_i}{e_i} = (G\Delta e - B\Delta f)_i + \frac{f_i}{e_i} (B\Delta e + G\Delta f)_i
\] (3.12)

\[
\frac{\Delta Q_i}{e_i} = -(B\Delta e + G\Delta f)_i - \frac{f_i}{e_i} (G\Delta e - B\Delta f)_i
\] (3.13)
3. Δe_i and Δf_i are very small and tend to zero as convergence is reached.

Therefore

\[ f_i (B \Delta e), f_i (G \Delta f), f_i (G \Delta e), \text{ and } f_i (B \Delta f) \text{ are negligible. Hence equations (3.12) and (3.13) can be further simplified as} \]

\[ \frac{\Delta P_i}{e_i} = (G \Delta e - B \Delta f)_i \tag{3.14} \]

\[ \frac{\Delta Q_i}{e_i} = -(B \Delta e + G \Delta f)_i \tag{3.15} \]

Equations (3.14) and (3.15) can now be combined as

\[
\begin{bmatrix}
\Delta P \\
\Delta Q
\end{bmatrix}
= 
\begin{bmatrix}
-B' & G' \\
G'^T & B''
\end{bmatrix}
\begin{bmatrix}
\Delta f \\
\Delta e
\end{bmatrix}
\tag{3.16}
\]

where \( B' \) is obtained from the \( B \) matrix by eliminating the row and column corresponding to the slack bus. \( B'' \) is obtained from the \( B' \) matrix by eliminating the row and column corresponding to the PV buses. \( G' \) is obtained from the \( G \) matrix by eliminating the columns corresponding to the PV buses. \( G'^T \) is obtained from the \( G \) matrix by eliminating the rows corresponding to the PV buses.

For the load flow, we have to solve the following equation iteratively [127]:

\[
\begin{bmatrix}
\Delta f \\
\Delta e
\end{bmatrix}
= 
\begin{bmatrix}
-B' & G'^T \\
G' & B''
\end{bmatrix}^{-1}
\begin{bmatrix}
\frac{\Delta P}{e} \\
\frac{\Delta Q}{e}
\end{bmatrix}
\tag{3.17}
\]
3.3 Relation between the Constant Matrix and the $Z_{bus}$

In equation (3.17) the first matrix on the right hand side is the inverse of the constant matrix and has to be evaluated only once, since the elements of the constant matrix have been made independent of the variables $e$ and $f$. The size of the constant matrix for a system containing only PQ buses is $(2N-2) \times (2N-2)$. For a system containing PV buses, the size becomes $(2N-NPV-2) \times (2N-NPV-2)$. Normally, a practical power system has a large PQ/PV bus ratio. So the size of the constant matrix is quite large. For obtaining a load flow solution this large size matrix has to be inverted once.

In the following Section it is proved that the elements of the inverse of the constant matrix can be obtained directly from the elements of the system bus impedance matrix ($Z_{bus}$). The size of the bus impedance matrix is only $(N-1) \times (N-1)$ and the time required to construct the bus impedance matrix is much less when compared with the time required to evaluate the inverse of the constant matrix of large size. To illustrate this, first consider only the PQ buses. We shall now prove that the elements of the inverse of the constant matrix are the elements of the bus impedance matrix.

Let $Y = G + jB$ and $Z = R + jX$, where $Y$ is the complex admittance matrix, and $Z$ is the complex impedance matrix. Then $YZ = I$. Hence,

$$(G + jB)(R + jX) = (GR - BX) + j(BR + GX) = I + j0$$

or

$$GR - BX = I \quad \text{and} \quad BR + GX = 0$$

Now let $H$ and $A$ be two matrices such that
\[ A = \begin{bmatrix} X & R \\ R & -X \end{bmatrix} \]  \hspace{1cm} (3.20)

\[ H = \begin{bmatrix} -B & G \\ G & B \end{bmatrix} \]

Then

\[ HA = \begin{bmatrix} -B & G \\ G & B \end{bmatrix} \begin{bmatrix} X & R \\ R & -X \end{bmatrix} = \begin{bmatrix} (-BX + GR) & (-BR - GX) \\ (GX + BR) & (GR - BX) \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} = I \]

where \( I \) is the identity matrix of order \((2N-2) \times (2N-2)\)

Hence,

\[ H^{-1} = A = \begin{bmatrix} X & R \\ R & -X \end{bmatrix} \]  \hspace{1cm} (3.22)

Thus, it is seen that the inverse of the constant matrix is composed of sub-matrices, which are the resistive and reactive submatrices of the system bus impedance matrix. Hence, it is now proved that the inverse of the constant matrix can be derived from the system bus impedance matrix \( Z_{\text{BUS}} \).

For systems with PV buses the following modifications have to be made:

When PV buses are also present in the systems, we have an additional equation for the voltage magnitude at the PV buses.

\[ V^2 = e^2 + f^2 \]  \hspace{1cm} (3.23)

Partial derivatives of \( V \) gives

\[ 0 = 2e \Delta e + 2f \Delta f \]  \hspace{1cm} (3.24)
Hence

\[ \Delta e = - \frac{f}{e} \Delta f \]  

(3.25)

Since \( f \) is normally small, \( \Delta e \) for a PV bus is also small. Therefore, it can be neglected.

Let the load flow equation be written as

\[
\begin{bmatrix}
\frac{\Delta P_i}{e_i} \\
\frac{\Delta Q_i}{e_i}
\end{bmatrix} = \begin{bmatrix} -B & G \\
G^T & B \end{bmatrix} \begin{bmatrix} \Delta f \\ \Delta e \end{bmatrix}
\]  

(3.26)

where \( \Delta f \) is a vector of all buses, but \( \Delta e \) is a vector of PQ buses only.

\( G' \) contains columns of PQ buses only taken from \( G \), and \( B' \) contains rows and columns of PQ buses only taken from \( B \). Since we have \([YZ] = I\), hence

\[
\begin{bmatrix} -B & G' \\
G'^T & B' \end{bmatrix}^{-1} \begin{bmatrix} X & R \\
R^T & -X \end{bmatrix} = I
\]  

(3.27)

Let the matrices on the left hand side be partitioned as below, where the elements of first row and column correspond to PQ and PV buses, second row and columns correspond to PQ buses only and third row and columns correspond to PV buses.

\[
\begin{bmatrix} -B & G_1 & G_2 \\
G_1^T & B_{11} & B_{12} \\
G_2^T & B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} X & R_1 & R_2 \\
R_1^T & -X_{11} & -X_{12} \\
R_2^T & -X_{21} & -X_{22} \end{bmatrix} = I
\]  

(3.28)

then

\[
\begin{bmatrix} X & R_1 & R_2 \\
R_1^T & -X_{11} & -X_{12} \\
R_2^T & -X_{21} & -X_{22} \end{bmatrix} = \begin{bmatrix} -B & G_1 & G_2 \\
G_1^T & B_{11} & B_{12} \\
G_2^T & B_{21} & B_{22} \end{bmatrix}^{-1}
\]  

(3.29)

The load flow equation (3.26) can be written as
\[
\begin{bmatrix}
\frac{\Delta P}{e} \\
\frac{-\Delta Q_1}{e} \\
\frac{-\Delta Q_2}{e}
\end{bmatrix} =
\begin{bmatrix}
-B & G_1 & G_2 \\
G_1^T & B_{11} & B_{12} \\
G_2^T & B_{21} & B_{22}
\end{bmatrix}
\begin{bmatrix}
\Delta f \\
\Delta e_1 \\
\Delta e_2
\end{bmatrix}
\] (3.30)

or
\[
\begin{bmatrix}
\Delta f \\
\Delta e_1 \\
\Delta e_2
\end{bmatrix} =
\begin{bmatrix}
-B & G_1 & G_2 \\
G_1^T & B_{11} & B_{12} \\
G_2^T & B_{21} & B_{22}
\end{bmatrix}^{-1}
\begin{bmatrix}
\frac{\Delta P}{e} \\
\frac{-\Delta Q_1}{e} \\
\frac{-\Delta Q_2}{e}
\end{bmatrix}
\] (3.31)

where \( \Delta e_1 \) and \( \Delta Q_1 \) are for PQ buses and \( \Delta e_2 \) and \( \Delta Q_2 \) are for PV buses. Combining equations (3.29) and (3.31), the following equation is obtained

\[
\begin{bmatrix}
\frac{\Delta f}{e} \\
\frac{\Delta e_1}{e} \\
\frac{\Delta e_2}{e}
\end{bmatrix} =
\begin{bmatrix}
X & R_1 & R_2 \\
R_1^T & -X_{11} & -X_{12} \\
R_2^T & -X_{21} & -X_{22}
\end{bmatrix}
\begin{bmatrix}
\frac{\Delta P}{e} \\
\frac{-\Delta Q_1}{e} \\
\frac{-\Delta Q_2}{e}
\end{bmatrix}
\] (3.32)

For simplification, let us write equation (3.32) as

\[
\begin{bmatrix}
\frac{\Delta f}{e} \\
\frac{\Delta e_1}{e} \\
\frac{\Delta e_2}{e}
\end{bmatrix} =
\begin{bmatrix}
X & R_1 & R_2 \\
R_1^T & -X_{11} & -X_{12} \\
R_2^T & -X_{21} & -X_{22}
\end{bmatrix}
\begin{bmatrix}
\frac{\Delta P}{e} \\
\frac{-\Delta Q_1}{e} \\
\frac{-\Delta Q_2}{e}
\end{bmatrix}
\] (3.32a)

For PV buses \( \Delta e_2 \) is approximately zero. Hence eliminating \( \Delta Q_2 \) from the above matrix equation (3.32a)

\[
0 = R_2^T \Delta P - X_{21} \Delta Q_1 - X_{22} \Delta Q_2
\] (3.33)

\[
\Delta Q_2 = X_{21}^{-1} R_2^T \Delta P - X_{22}^{-1} X_{21} \Delta Q_1
\] (3.34)
Substituting $\Delta Q_2$ for $\Delta Q_2$ in the equations of $\Delta f$ and $\Delta e_1$

$$\Delta f = X\Delta P + R_1\Delta Q_1 + R_2X_{22}^{-1}R_2^T\Delta P - R_2X_{22}^{-1}X_{21}\Delta Q_1$$  \hfill (3.35)

$$\Delta f = (X + R_2X_{22}^{-1}R_2^T)\Delta P + (R_1 - R_2X_{22}^{-1}X_{21})\Delta Q_1$$  \hfill (3.36)

$$\Delta e_1 = R_1^T\Delta P - X_{11}\Delta Q_1 - X_{12}X_{22}^{-1}R_2^T\Delta P + X_{12}X_{22}^{-1}X_{21}\Delta Q_1$$  \hfill (3.37)

$$\Delta e_1 = (R_1 - X_{12}X_{22}^{-1}R_2^T)\Delta P - (X_{11} - X_{12}X_{22}^{-1}X_{21})\Delta Q_1$$  \hfill (3.38)

From the expressions derived above it is now proved that

$$\begin{bmatrix} -B & G^T \\ G^T & B^T \end{bmatrix}^{-1} = \begin{bmatrix} (X + R_2X_{22}^{-1}R_2^T) & (R_1 - R_2X_{22}^{-1}X_{21}) \\ (R_1 - X_{12}X_{22}^{-1}R_2^T) & (-X_{11} + X_{12}X_{22}^{-1}X_{21}) \end{bmatrix}$$  \hfill (3.39)

Hence, if PV buses are present in the system, the inverse of the constant matrix can still be computed from $Z_{BUS}$ using equation (3.39). It is shown in Table - 3.14 that even after these modifications for PV buses, which are very few in number, the overall time required for the inversion of a constant matrix is much larger than the building time of the $Z_{BUS}$. The inversion time will increase linearly with the increase in the system size. Hence, it is economical to use $Z_{BUS}$ matrix instead of inverting the constant matrix.

### 3.4 System Studies

The proposed constant matrix load flow algorithm in rectangular coordinates is tested on 5-Bus system [165], IEEE 14-Bus test system and IEEE 30-Bus test system. The proposed algorithm is compared with the FDLF model in polar and rectangular coordinates and a constant matrix model in polar coordinates. Tables - 3.1, 3.2 and 3.3 shows the number of iterations and iteration time required for 5-bus, 14-bus and 30-bus systems, respectively. Tables - 3.4, 3.5 and 3.6 shows the total time required for convergence for the above systems. Number of iterations required for convergence with varying number of PV buses in IEEE
14-Bus and IEEE 30-Bus systems are given in Table 3.7. Effect of varying load on the number of iterations required are shown in Table 3.8. The proposed algorithm is also tested for capacitive series branches. Table 3.9 shows the effect of capacitive series branches on the number of iterations required for 5-bus system and Table 3.10 shows the number of success and failures for 14 and 30-bus system when individual lines are made capacitive. Effect of change in transformer taps on convergence are given in Table 3.11. Table 3.13 gives the number of iterations required if multiple lines are made capacitive in 14-bus system. Time required to construct the $Z_{BUS}$ and to invert the constant matrix for all the systems are given in Table 3.13. Effect of changes in the number of the PV buses on the $Z_{BUS}$ building time and the time required for the invert the constant matrix are shown in Table 3.14.

**Table - 3.1**

**Number of iterations and iterations time for 5-bus system**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Time</td>
<td>Time</td>
<td>Time</td>
</tr>
<tr>
<td>1.0</td>
<td>5 0.11</td>
<td>5 0.11</td>
<td>7 0.16</td>
<td>5 0.06</td>
</tr>
<tr>
<td>1.5</td>
<td>7 0.17</td>
<td>7 0.13</td>
<td>6 0.14</td>
<td>5 0.06</td>
</tr>
<tr>
<td>2.0</td>
<td>10 0.21</td>
<td>11 0.22</td>
<td>6 0.14</td>
<td>5 0.06</td>
</tr>
<tr>
<td>2.5</td>
<td>20 0.53</td>
<td>23 0.55</td>
<td>7 0.16</td>
<td>5 0.06</td>
</tr>
</tbody>
</table>
### Table - 3.2

**Number of iterations and iterations time for 14-bus system**

<table>
<thead>
<tr>
<th>Rline x</th>
<th>FDLF Polar</th>
<th>FDLF Rectangular</th>
<th>Const. Jac. Polar</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Time</td>
<td>Time</td>
<td>Time</td>
</tr>
<tr>
<td>1.0</td>
<td>18</td>
<td>16</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>2.96</td>
<td>2.47</td>
<td>1.10</td>
<td>0.88</td>
</tr>
<tr>
<td>1.5</td>
<td>Fail</td>
<td>Fail</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>----</td>
<td>----</td>
<td>1.21</td>
<td>0.88</td>
</tr>
<tr>
<td>2.0</td>
<td>Fail</td>
<td>Fail</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>----</td>
<td>----</td>
<td>1.21</td>
<td>0.88</td>
</tr>
<tr>
<td>2.5</td>
<td>Fail</td>
<td>Fail</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>----</td>
<td>----</td>
<td>1.53</td>
<td>1.37</td>
</tr>
</tbody>
</table>

### Table - 3.3

**Number of iterations and iteration time for 30-bus system**

<table>
<thead>
<tr>
<th>Rline x</th>
<th>FDLF Polar</th>
<th>FDLF Rectangular</th>
<th>Const. Jac. Polar</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Time</td>
<td>Time</td>
<td>Time</td>
</tr>
<tr>
<td>1.0</td>
<td>17</td>
<td>16</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>11.75</td>
<td>11.10</td>
<td>5.93</td>
<td>4.56</td>
</tr>
<tr>
<td>1.5</td>
<td>Fail</td>
<td>Fail</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>----</td>
<td>----</td>
<td>5.93</td>
<td>4.56</td>
</tr>
<tr>
<td>2.0</td>
<td>Fail</td>
<td>Fail</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>----</td>
<td>----</td>
<td>6.54</td>
<td>5.16</td>
</tr>
<tr>
<td>2.5</td>
<td>Fail</td>
<td>Fail</td>
<td>16</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>----</td>
<td>----</td>
<td>9.50</td>
<td>8.57</td>
</tr>
</tbody>
</table>

### Table - 3.4

**Total solution time required for 5-bus system**

<table>
<thead>
<tr>
<th>Rline x</th>
<th>FDLF Polar</th>
<th>FDLF Rectangular</th>
<th>Const. Jac. Polar</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.16</td>
<td>0.17</td>
<td>0.27</td>
<td>0.10</td>
</tr>
<tr>
<td>1.5</td>
<td>0.22</td>
<td>0.22</td>
<td>0.22</td>
<td>0.10</td>
</tr>
<tr>
<td>2.0</td>
<td>0.28</td>
<td>0.33</td>
<td>0.22</td>
<td>0.10</td>
</tr>
<tr>
<td>2.5</td>
<td>0.60</td>
<td>0.65</td>
<td>0.27</td>
<td>0.10</td>
</tr>
</tbody>
</table>
### Table - 3.5
Total solution time required for 14-bus system

<table>
<thead>
<tr>
<th>Rline x</th>
<th>FDLF Polar</th>
<th>FDLF Rectangular</th>
<th>Const. Jac. Polar</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>3.57</td>
<td>3.02</td>
<td>2.80</td>
<td>1.32</td>
</tr>
<tr>
<td>1.5</td>
<td>Fail</td>
<td>Fail</td>
<td>2.91</td>
<td>1.32</td>
</tr>
<tr>
<td>2.0</td>
<td>Fail</td>
<td>Fail</td>
<td>2.91</td>
<td>1.32</td>
</tr>
<tr>
<td>2.5</td>
<td>Fail</td>
<td>Fail</td>
<td>3.24</td>
<td>1.81</td>
</tr>
</tbody>
</table>

### Table - 3.6
Total solution time required for 30-bus system

<table>
<thead>
<tr>
<th>Rline x</th>
<th>FDLF Polar</th>
<th>FDLF Rectangular</th>
<th>Const. Jac. Polar</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>17.68</td>
<td>16.97</td>
<td>28.56</td>
<td>7.19</td>
</tr>
<tr>
<td>1.5</td>
<td>Fail</td>
<td>Fail</td>
<td>28.56</td>
<td>7.19</td>
</tr>
<tr>
<td>2.0</td>
<td>Fail</td>
<td>Fail</td>
<td>29.17</td>
<td>7.79</td>
</tr>
<tr>
<td>2.5</td>
<td>Fail</td>
<td>Fail</td>
<td>32.13</td>
<td>11.20</td>
</tr>
</tbody>
</table>

### Table - 3.7
Number of iterations required with varying PV buses

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>19</td>
<td>9</td>
<td>16</td>
<td>16</td>
<td>11</td>
<td>15</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>19</td>
<td>10</td>
<td>16</td>
<td>16</td>
<td>10</td>
<td>11</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>11</td>
<td>16</td>
<td>16</td>
<td>9</td>
<td>10</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>18</td>
<td>13</td>
<td>16</td>
<td>16</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>18</td>
<td>14</td>
<td>16</td>
<td>15</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>
### Table - 3.8
Number of iterations required with varying loading conditions

<table>
<thead>
<tr>
<th>Loading ↓</th>
<th>Test Systems→</th>
<th>FDLF (Polar)</th>
<th>FDLF (Rect.)</th>
<th>Cons. Jac. (Polar)</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>14</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>P + j Q</td>
<td>5</td>
<td>18</td>
<td>17</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2 P + j Q</td>
<td>5</td>
<td>18</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>P + j 2 Q</td>
<td>4</td>
<td>18</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>0.5 P + j 0.5 Q</td>
<td>6</td>
<td>17</td>
<td>18</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>0.2 P + j 0.2 Q</td>
<td>6</td>
<td>10</td>
<td>17</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2 P + j 2 Q</td>
<td>5</td>
<td>18</td>
<td>13</td>
<td>8</td>
</tr>
</tbody>
</table>

### Table - 3.9
Iterations required for 5-bus system when individual lines made capacitive

<table>
<thead>
<tr>
<th>Line nos.→</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Success</th>
<th>Failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm↓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FDLF(Polar)</td>
<td>5</td>
<td>14</td>
<td>6</td>
<td>Fail</td>
<td>12</td>
<td>Fail</td>
<td>Fail</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>FDLF(Rect.)</td>
<td>5</td>
<td>27</td>
<td>6</td>
<td>Fail</td>
<td>12</td>
<td>Fail</td>
<td>Fail</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Const. Jac.</td>
<td>2</td>
<td>10</td>
<td>7</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Proposed</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### Table - 3.10
Number of successes and failures when individual lines are made capacitive

<table>
<thead>
<tr>
<th>Test System →</th>
<th>IEEE 14-Bus</th>
<th>IEEE 30 -Bus</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Success</td>
<td>Failures</td>
</tr>
<tr>
<td>Algorithm ‡</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FDLF(Polar)</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>FDLF(Rectangular)</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>Const. Jac.</td>
<td>17</td>
<td>3</td>
</tr>
<tr>
<td>Proposed</td>
<td>17</td>
<td>3</td>
</tr>
</tbody>
</table>

### Table - 3.11
Number of iterations required after varying the tap of the transformer in line 8 of 14-bus and line 11 of 30-bus system

<table>
<thead>
<tr>
<th>Transformer Tap</th>
<th>FDLF Polar</th>
<th>FDLF Rectangular</th>
<th>Const. Jac. Polar</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>14-bus 30-bus</td>
<td>14-bus 30-bus</td>
<td>14-bus 30-bus</td>
<td>14-bus 30-bus</td>
</tr>
<tr>
<td>0.9</td>
<td>18</td>
<td>18</td>
<td>16</td>
<td>9 10</td>
</tr>
<tr>
<td>1.0</td>
<td>18</td>
<td>17</td>
<td>16</td>
<td>9 10</td>
</tr>
<tr>
<td>1.1</td>
<td>18</td>
<td>16</td>
<td>16</td>
<td>10 10</td>
</tr>
</tbody>
</table>

### Table - 3.12
Number of iterations required when multiple lines are made capacitive in 14-bus example

<table>
<thead>
<tr>
<th>Lines made capacitive</th>
<th>FDLF Polar</th>
<th>FDLF Rectangular</th>
<th>Const. Jac. Polar</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>2, 14</td>
<td>Fail</td>
<td>Fail</td>
<td>Fail</td>
<td>Fail</td>
</tr>
<tr>
<td>8, 13</td>
<td>19</td>
<td>16</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>7, 13</td>
<td>17</td>
<td>16</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>17, 18</td>
<td>Fail</td>
<td>Fail</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>8, 11</td>
<td>19</td>
<td>16</td>
<td>7</td>
<td>6</td>
</tr>
</tbody>
</table>
### Table - 3.13
Comparison of time required to build $Z_{bus}$ and invert constant matrix

<table>
<thead>
<tr>
<th>Test System</th>
<th>$Z_{bus}$ Building Time (Seconds)</th>
<th>Time required for the Inversion of Constant matrix (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Bus</td>
<td>0.06</td>
<td>01.62</td>
</tr>
<tr>
<td>IEEE 14-Bus</td>
<td>0.11</td>
<td>02.63</td>
</tr>
<tr>
<td>24-Bus</td>
<td>0.33</td>
<td>14.57</td>
</tr>
<tr>
<td>IEEE 30-Bus</td>
<td>0.55</td>
<td>29.06</td>
</tr>
</tbody>
</table>

### Table - 3.14
Comparison of time required to build $Z_{bus}$ and invert constant matrix with varying PV buses

<table>
<thead>
<tr>
<th>Test System</th>
<th>One PV bus</th>
<th>Two PV buses</th>
<th>Three PV buses</th>
<th>Four PV buses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>$Z_{bus}$</td>
<td>Inverse</td>
<td>$Z_{bus}$</td>
<td>Inverse</td>
</tr>
<tr>
<td>IEEE 14-Bus</td>
<td>0.28</td>
<td>02.42</td>
<td>0.33</td>
<td>02.09</td>
</tr>
<tr>
<td>24-Bus</td>
<td>0.83</td>
<td>13.68</td>
<td>1.04</td>
<td>12.74</td>
</tr>
<tr>
<td>IEEE 30-Bus</td>
<td>1.31</td>
<td>27.57</td>
<td>1.65</td>
<td>26.14</td>
</tr>
</tbody>
</table>

### 3.5 Discussion

#### 3.5.1 Core Storage

As the elements of the inverse of the constant matrix can be directly calculated from the $Z_{bus}$ of size $(N-1) \times (N-1)$, there is no need to store the entire matrix of size $(2N-2) \times (2N-2)$. Hence the core storage requirement is nearly the same as that of the FDLF method; in contrast the memory requirements for the second order load flow model of Sachdev et al. [146] is much more as it stores the second order terms also.
3.5.2 PV-Bus Q-limits

When the load flow solution has moderately converged, any Q-limit violations at PV buses should be corrected. The affected PV bus is converted to the PQ type with the reactive generation set at the limiting value. The voltage of the converted bus is subsequently compared to the scheduled value and the bus is re-converted to the PV type, if any of the back-off conditions are satisfied. Switching of bus types, in the case of the FDLF model is simulated by the insertion/deletion of rows and columns in the matrix \([B']\) and, hence, requires its re-triangularization. Each affected PV bus is converted to the PQ type with a corresponding modification of the matrix \([B']\). Hence, each violation of the Q-limit causes modification of \([B']\) and thus requires fresh inversion. In the proposed method, when the Q-limit is violated then the elements of the new constant matrix are still obtained from the \(Z_{bus}\) by modifying the \(Z_{bus}\) using a simple formula already derived in this Chapter.

3.5.3 Effect of addition and removal of lines

It is also observed that if the resistance or the reactance of any line is changed, or a new line is added to the system, or removed from the system, then calculating the inverse of the modified constant matrix from the original matrix is a trivial job. It is well known that the existing bus impedance matrix can be easily modified (Section 2.6) for corresponding system changes. The time required for modifying the bus impedance matrix is much less than the total time required to construct a new bus admittance matrix and then calculating its inverse.

3.5.4 Performance in the presence of capacitive series branches

The proposed algorithm has also been tested for capacitive series branches. When the lines are made capacitive, successively, one by one, it is found that the total number of successes in convergence is much more with the proposed algorithm as compared to others. Even if more than one line are simultaneously made capacitive, the proposed algorithm works
better. Whenever a line is made capacitive, the constant matrix will change and a fresh inversion is required. However, the inverse of the modified constant matrix can be easily obtained from the $Z_{BUS}$ using a simple procedure explained in Chapter II (Section 2.6).

3.5.5 Effect of varying bus loading

The real and reactive power loads on the buses are changed from 20% to 200% of the normal loading and it is found that for most of the loading conditions the proposed algorithm requires lesser time as compared to the others.

3.5.6 Solution Time

The solution time required by the proposed algorithm is 37.0% of its counterpart in polar coordinates, 58.8% of the FDLF (rectangular) and 62.5% of FDLF (polar) method for a 5-bus system. For a 14-bus system it requires 47.14%, 43.70% and 36.97% respectively. And for a 30-bus system it requires 25.13%, 42.36% and 40.66% respectively. The saving in time is mainly due to the avoidance of an inversion and the effectiveness of the load flow model.

3.6 Conclusions

A constant matrix load flow model in rectangular coordinates is discussed.

In this Chapter, it has been shown that the inverse of the constant matrix can be easily derived from the $Z_{BUS}$ of the system. This speeds up the method considerably as the inverse of the constant matrix is a major step in the computations. Further, this reduces the storage requirements because only $Z_{BUS}$ needs to be stored now.

The inverse of the new constant matrix can also be easily obtained from the stored $Z_{BUS}$ for PV bus switching during the iterative process, by modifying the elements of $Z_{BUS}$ using the expressions derived in Section 3.3. The overall solution time thus reduces mainly due to the fact that, when the Q-limit violations take place the constant matrix has to be modified and a fresh inversion is required, which is obtained directly from the $Z_{BUS}$.

Since the existing $Z_{BUS}$ can be easily modified for simple system modifications, e.g.,
addition or deletion of a line or switching of capacitive series branches, the inverse of the constant matrix of the modified system can be easily derived from the $Z_{\text{bus}}$. The performance of the constant matrix load flow has been compared with its polar coordinate version and the FDLF models. From the results obtained from various tests performed on a 5-bus, IEEE 14-bus and IEEE 30-bus test system the following conclusions have been drawn:

The proposed constant matrix load flow model:

- requires lesser storage as compared to other methods, since storing the $Z_{\text{bus}}$ is enough to generate the inverse of the constant matrix.
- converges for many types of ill-conditioned systems where the FDLF model fails.
- has faster convergence properties as compared to other algorithms even when PV bus switching takes place. It also exhibits better convergence characteristics in the presence of single and multiple capacitive series branches. The number of iterations required are less for different bus loading conditions and changed transformer taps.