APPLICATION OF KRYLOV SUBSPACE METHODS IN POWER SYSTEMS

A Project Report

submitted by

ARUN L

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

MAY 2013

CERTIFICATE

This is to certify that the thesis titled **APPLICATION OF KRYLOV SUBSPACE METHODS IN POWER SYSTEMS**, submitted by **ARUN L**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bona fide record of the research work done by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. S Krishna
(Project Guide)
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 6th May 2013

ACKNOWLEDGEMENTS

It gives me great pleasure in expressing my sincere gratitude to my project guide Dr. S Krishna for giving me an opportunity to work in this area and for his excellent guidance, motivation and constant support throughout my project. I would like to express my gratitude to Prof. Enakshi Bhattacharya, Head of the Department of Electrical Engineering, IIT Madras.

I would like to express my heartfelt gratitude to research scholars in our lab Mr. Sunil S. Damodhar, Mr. Tumbali S. S. K. Chaitanya and Mr. Sarojkumar K for their valuable suggestions and support throughout my work.

With Great pleasure and respect, I express my utmost gratitude to my beloved parents Mr. Louis G and Mrs. Lillykutty P and my brother Anish L for their constant encouragement and support. Above all, I thank God Almighty for his guidance and tremendous blessings bestowed throughout my life.

ABSTRACT

KEYWORDS: Krylov Subspace; Power Flow; Arnoldi's Method; Small Signal Stability.

Solution of large sparse linear system of equations form the core of power system computations, whether it is state estimation or security assessment. Static security assessment of power systems is a time consuming task involving repetitive solution of power flow equations. There is a need to speed up this process by improved numerical algorithms which can substantially reduce the computational time.

Krylov Subspace Power Flow (KSPF) uses a new approach based on the solution of large sparse linear system of equations using projection process onto Krylov subspaces. In this work the performance of KSPF is compared with that of the conventional Newton-Raphson power flow method. It includes three different implementation alternatives and the numerical results are obtained.

The small signal stability assessment of power systems requires evaluation of eigenvalues of a very large unsymmetric matrix. In large interconnected power systems fast eigenvalue computation is necessary for online oscillatory stability assessment. Arnoldi's method can be used to compute the eigenvalues of interest by doing an eigenvalue analysis of an upper Hessenberg matrix of lower dimension. This work discuss the theory of selective eigenvalue computation using Arnoldi's method and relevant numerical results obtained. A sparsity oriented implementation is also included, which avoids the direct computation of state matrix. The performance of Arnoldi's method is compared with the QR method and other selective eigenvalue computation techniques.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

KSPF	Krylov Subspace Power Flow
AESOPS	Analysis of Essentially Spontaneous Oscillations in Power Systems

NOTATIONS

$\|v\|$	Euclidean Norm of vector v
$\langle , \rangle$	Dot Product
$\mathbb{R}^N$	N Dimensional Euclidean Space
$\perp$	Orthogonal
P^{sp}	Specified Active Power
Q^{sp}	Specified Reactive Power
δ	Generator Rotor Angle
ω	Generator Rotor Speed
D	Damping Coefficient
H	Generator Inertia Constant
$X_d^{'}$	Generator Direct Axis Transient Reactance

CHAPTER 1

Introduction

Secure operation of interconnected power systems has always been a challenging task because of their intrinsic dynamic nature; load, generation, network topology and key operating parameters are constantly changing. While security is primarily a function of the physical system and its current attributes, secure operation is facilitated by availability of adequate analytical tools for online computation.

1.1 Online Computation

Online computation for power system security can be broken down into three major functions that are carried out in an operations control center:

1. System monitoring
2. Contingency analysis
3. Security constrained optimal power flow.

System monitoring provides the operators of the power system with pertinent up to date information on the conditions of the power system. It involves systems for measurement and data transmission. State estimation is often used in such systems to combine the collected system data with system models to produce the best estimate of the current power system conditions or state.

The results of contingency analysis allow systems to be operated defensively. Many of the problems that occur on a power system can cause serious trouble within such a quick period that the operator could not take action fast enough. This is often the case with cascading failures. Contingency programs based on a model of the power system are used to study outage events and alarm the operators to any potential over loads or out of limit voltages.

In security constrained optimal power flow, a contingency analysis is combined with an optimal power flow which seeks to make changes to the optimal dispatch of

generation, as well as other adjustments, so that when a security analysis is run, no contingencies result in violations.

Online computation for system security can be broadly divided into two,
1. Static Security Assessment
2. Dynamic Security Assessment
which are discussed below.

1.1.1 Static Security Assessment

Static security of a power system is defined as the ability of the system, following a contingency, to reach an operating state within the specified safety and supply quality. The assessment is based on the fact that the fast acting automatic control devices have restored the generation-load balance, but the slow acting controls and human decisions have not yet responded.

The static security assessment of a large power system is a computationally demanding task. It involves the solution of several nonlinear models (AC power flow) containing a large number of variables and constraints that define the feasible region of operation. In addition, the amount of memory required to store the massive data for different system configurations and contingencies is equally prohibitive. These considerations seriously undermine the application of static security assessment, in real time, without the support of large computing capability.

1.1.2 Dynamic Security Assessment

Online dynamic security assessment is a major concern in the operation of modern power systems. With the ever growing demand of elecricity, electrical plants are enforced to generate electrical conditions nearer to their security limits. Under these conditions, any large disturbance could jeopardize the system security and may lead to cascading outages and a shut down of partial or major portion of the power system. In order for preventive action to be taken and to alert system operator in case a need arises, a fast online security tool for analyzing the level of security of power system is imperative to be developed.

Online dynamic security assessment mainly focus on the evaluation of the ability of a power system to maintain synchronism under severe but plausible contingencies. A contingency is a large disturbance, which could be a sudden loss of load, significant changes in the network or a severe fault such as three phase short circuit at the generator bus.

1.2 Literature Review

A real time dynamic security assessment and early warning system for the purpose of keeping interconnected power systems operating in security region, monitoring dynamic security margin constantly and ensuring system operation reliability is proposed in [1]. A generation rescheduling method to increase the dynamic security of power systems based on the idea of coherent behavior of generators is proposed in [2]. An online dynamic security assessment scheme for large scale interconnected power systems using phasor measurements and decision trees is proposed in [3]. A class of intelligent algorithms for online dynamic security assessment are proposed in [4].

The Krylov subspace methods [5] have been developed and perfected, starting approximately in the early 1980's , for the iterative solution of the linear problem $Ax = b$ for large sparse non symmetric matrices. GMRES [6] is the most widely known and used Krylov subspace method.

The evaluation of small signal stability analysis of power systems requires the calculation of the eigenvalues of a very large unsymmetric non-sparse matrix. The conventional QR algorithm calculates all the eigenvalues to determine the small signal stability of power systems. Poorly damped electromechanical oscillations have become a common phenomenon in the well developed power systems and this small signal stability problem as well as the voltage related problems become one of the most important factors to place a limit on power transfer capability and to jeopardize safe operation [7]. Fast eigenvalue computation in large interconnected power systems is useful for online oscillatory stability assessment.

Many efficient techniques for the small signal stability analysis of large power systems have been developed in the last two decades that focus on evaluating a selected subset of eigenvalues associated with the complete system response. The AESOPS pro-

gram computes an electromechanical oscillation mode of interest based on a good initial guess [8]. Newton's method can also be used for selecively computing the eigenvalues of interest. The selective modal analysis approach [9] computes eigenvalues of interest by constructing a reduced order model that involves variables relevant to the selected modes .

1.3 Motivation and Objectives

Phenomena that can compromise power systems operation need to be carefully analyzed in order to evaluate their impact on the security and reliability levels of the electrical networks. Security analysis of a power system is a time-intensive task. There is a need to speed up the online computations for real time assessment of security and stability levels of power systems such that corrective actions can be taken in time.

Solution of large sparse linear system of equations form the core of power system computations whether it is power flow, state estimation or security assessment [10]. Power flow programs are probably the most fundamental and widely used tools in the analysis of power systems. Repetitive solution of powerflow equations are necessary for the static security assessment of a power system. The Krylov subspace methods [5] have been developed and perfected, starting approximately in the early 1980's , for the iterative solution of the linear problem $Ax = b$ for large sparse non symmetric matrices. The power flow problem is however nonlinear and this work evaluates the performance of Krylov subspace methods when extended to the solution of nonlinear equations [11].

The evaluation of small signal stability analysis of power systems requires the calculation of the eigenvalues of a very large unsymmetric non-sparse matrix. The conventional QR algorithm calculates all the eigenvalues to determine the small signal stability of power systems. Arnoldi's method can be used to compute the eigenvalues close to a complex shift point [12]. The present work compares the performance of Arnoldi's method in selectively computing the eigenvalues of interest with other existing methods.

Objectives of this work can be enlisted as follows

1. To Evaluate the performance of KSPF in comparison with conventional Newton-Raphson power flow, when applied to standard test systems.
2. To compare the performance of Arnoldi's method in selectively computing the eigenvalues of a large power system with other existing methods.

1.4 Organization of the Report

Chapter 2 introduces the basic definitions and concepts of Krylov subspaces and Arnoldi's method.

In **chapter 3** the application of Krylov subspace methods in the solution of power flow problem and the numerical results obtained for standard test systems are presented.

Chapter 4 deals with the existing selective eigenvalue computation techniques in the literature such as Newton's method and AESOPS method and associated numerical results obtained on a 493 bus power system.

Chapter 5 presents the application of Arnoldi's method in selective computation of eigenvalues for the small signal stability analysis of a large power system. It also includes the numerical results obtained on a 493 bus power system.

The report is concluded in **chapter 6** with a summary of the work done and suggestions for future work.

CHAPTER 2

Krylov Subspace and Arnoldi's Method

Large sparse linear systems of equations or large sparse eigenvalue problems appear in most applications of scientific computing. In case the original problem is nonlinear, linearization by Newton's method or a Newton-type method again leads to a linear problem. Sparse linear systems of equations can be solved by either so-called direct solvers, which are clever variations of Gaussian elimination, or iterative methods [13]. Krylov subspace methods are a class of subspace iteration methods based on the projection to a Krylov subpace. Basic definitions associated with the Krylov subpace method are given below.

2.1 Krylov Subspace

Krylov subspace generated by an $N \times N$ matrix A and an $N \times 1$ vector b is the subspace spanned by the vectors of the Krylov sequence represented as

$$K = \text{span}\{b, Ab, A^2b, \ldots, A^{m-1}b\}$$

where m is a positive integer less than or equal to N.

From the above definition it directly follows that Krylov subspace is the subspace of all vectors x in $\mathbb{R}^N$ which can be written as $x = p(A)b$, where p is a polynomial of degree not exceeding $m - 1$.

2.1.1 Basis for Krylov Subspace K

In any inner product space it is free to choose the basis in which to work. It often greatly simplifies the calculations to work in an orthogonal basis. For example if $S = \{v_1, v_2, v_3, \ldots, v_N\}$ is an orthogonal basis for V then it is easy to express any vector

$w \in V$ as a linear combination of vectors in S.

$$w = \frac{\langle w, v_1 \rangle}{\|v_1\|^2} v_1 + \frac{\langle w, v_2 \rangle}{\|v_2\|^2} v_2 + \cdots + \frac{\langle w, v_N \rangle}{\|v_N\|^2} v_N$$

If $r_0 = b - Ax$ is proportional to an eigenvector of A, the vectors in the Krylov sequence $[r_0, Ar_0, A^2 r_0, ...A^{m-1} r_0]$ may be linearly dependent since $A^k r_0$ approaches the dominant eigenvector of A as k increases. Hence an orthogonal basis is needed. In KSPF method, Arnoldi's algorithm is used to generate an orthonormal basis for the Krylov subspace. Arnoldi's method uses a modified Gram-Schmidt method to generate an orthonormal basis.

Given an arbitrary basis $\{u_1, u_2, u_3 \ldots, u_N\}$ for an N dimensional inner product space V, Gram-Schmidt algorithm constructs an orthogonal basis $\{v_1, v_2, \ldots, v_N\}$. To get an orthonormal basis normalize each vector in the orthogonal basis.

2.2 Arnoldi's Method

If K=span$\{v, Av, A^2 v \ldots, A^{m-1} v\}$ is the given Krylov subspace and K_m is a matrix with columns $v, Av, A^2 v, \ldots, A^{m-1} v$, then the Arnoldi's algorithm [14] computes an orthonormal basis using modified Gram-Schmidt process. Modified Gram-Schmidt method need only the starting vector v_1 and the matrix-vector product evaluations, without explicitly using other vectors in the given basis.

Let v_1 be the first vector in the given basis and is chosen with a norm 1. The second vector in the given basis, Av_1 has to be orthogonalized against v_1. Let w_1 be the component of Av_1 orthogonal to v_1, which is obtained by subtracting the projection of Av_1 in the direction of v_1 from Av_1.

$$\begin{aligned} w_1 &= Av_1 - \frac{\langle Av_1, v_1 \rangle}{\|v_1\|^2} v_1 \\ &= Av_1 - \langle Av_1, v_1 \rangle v_1 \qquad (\because \|v_1\| = 1) \\ &= Av_1 - h_{11} v_1, \quad \text{where} \quad h_{11} = \langle Av_1, v_1 \rangle \end{aligned}$$

Normalizing this vector gives the second vector in the orthonormal basis as

$$\begin{aligned} v_2 &= \frac{w_1}{\|w_1\|} \\ &= \frac{w_1}{h_{21}}, \quad \text{where } h_{21} = \|w_1\| \end{aligned}$$

Based on modified Gram-Schmidt method Av_2 is taken as the new vector to be orthogonalized against the previous two vectors v_1 and v_2. Component of Av_2 orthogonal to both v_1 and v_2 is ,

$$\begin{aligned} w_2 &= Av_2 - \frac{\langle Av_2, v_1\rangle}{\|v_1\|^2}v_1 - \frac{\langle Av_2, v_2\rangle}{\|v_2\|^2}v_2 \\ &= Av_2 - \langle Av_2, v_1\rangle v_1 - \langle Av_2, v_2\rangle v_2 \\ &= Av_2 - h_{12}v_1 - h_{22}v_2 \end{aligned}$$

Normalizing this vector gives the third vector in the orthonormal basis as

$$\begin{aligned} v_3 &= \frac{w_2}{\|w_2\|} \\ &= \frac{w_2}{h_{32}} \end{aligned}$$

Repeating the above procedure, the m^{th} step will be

$$\begin{aligned} w_m &= Av_m - \frac{\langle Av_m, v_1\rangle}{\|v_1\|^2}v_1 - \frac{\langle Av_m, v_2\rangle}{\|v_2\|^2}v_2 \ldots - \frac{\langle Av_m, v_m\rangle}{\|v_m\|^2}v_m \\ &= Av_m - h_{m1}v_1 - h_{m2}v_2 - \ldots - h_{mm}v_m \\ v_{m+1} &= \frac{w_m}{\|w_m\|} \\ &= \frac{w_m}{h_{m+1,m}} \end{aligned}$$

The above procedure can be formulated as an algorithm as given below.

2.2.1 Arnoldi's Algorithm

Arnoldi's algorithm consists of the following steps.

1. Choose a vector v_1 of norm 1
2. For $j = 1, 2 \ldots, m$ Do:
3. Compute $h_{ij} = \langle Av_j, v_i \rangle$ for $i = 1, 2 \ldots, j$
4. Compute $w_j = Av_j - \sum_{i=1}^{j} h_{ij} v_i$
5. $h_{j+1,j} = \|w_j\|_2$
6. If $h_{j+1,j} = 0$ then stop
7. $v_{j+1} = w_j / h_{j+1,j}$
8. End Do

At each step, the Arnoldi's algorithm multiplies the previous Arnoldi vector v_j by A and then orthonormalizes the resulting vector w_j against all the previous vectors, v_i's by a modified Gram-Schmidt procedure. The special feature of Arnoldi's method is that the new vector to be orthogonalized is not taken from the columns of matrix K_m, but is expressed as Av_j, in terms of the last vector, v_j brought into the basis.

CHAPTER 3

Solution of Power Flow Equations Using Krylov Subspace Methodology

3.1 General Methodology

The power flow equations can be written in the general form

$$f(x) = 0 \tag{3.1}$$

where $x \in \mathbb{R}^N$ is the vector of voltage magnitudes and phase angles. $f(.)$ is the difference between the specified and calculated powers, both real and reactive. At k^{th} iteration the residual vector is given by

$$r_k = f(x^{(k)}) \tag{3.2}$$

where,

$$f(x^{(k)}) = \begin{bmatrix} f_1(x^{(k)}) \\ f_2(x^{(k)}) \\ \vdots \\ f_N(x^{(k)}) \end{bmatrix}, \; x^{(k)} = \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \cdots \\ x_N^{(k)} \end{bmatrix}$$

The linearized form of (3.2) is given by

$$\begin{aligned} r_k &= f(x^{(k)}) \\ &= f(x^{(k-1)} + h) \\ &= f(x^{(k-1)}) + J(x^{(k-1)})h \\ &= f(x^{(k-1)}) + J(x^{(k-1)})(x^{(k)} - x^{(k-1)}) \\ &= f(x^{(k-1)}) - J(x^{(k-1)})x^{(k-1)} + J(x^{(k-1)})x^{(k)} \end{aligned} \tag{3.3}$$

where,

$$J(x^{(k-1)}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial x_1} & \frac{\partial f_N}{\partial x_2} & \cdots & \frac{\partial f_N}{\partial x_N} \end{bmatrix} \Bigg|_{x=x^{(k-1)}}$$

In general form,

$$r_k = b_k - A_k x^{(k)} \tag{3.4}$$

Comparing (3.3) and (3.4) gives

$$A_k = -J(x_{k-1})$$

Here $-J(x_{k-1})$ is equal to the load flow Jacobian matrix in Newton-Raphson power flow method. Both (3.2) and (3.3) are used to obtain the minimal residual solution of the problem (3.1). Equation (3.3) is used to obtain a Krylov subspace update Δx_k for x_k and (3.2) to get a new minimum residual vector so that x_k is driven to the true solution. Let $x^{(0)}$ be the initial guess to the solution of (3.1), then (3.4) gives $r_0 = b - Ax^{(0)}$

Let

$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)} \tag{3.5}$$

be the new value of the vector of unknowns.

From (3.4),

$$\begin{aligned} r_{k+1} &= b_{k+1} - A_{k+1}x^{(k+1)} \\ &= b_{k+1} - A_{k+1}(x^{(k)} + \Delta x^{(k)}) \\ &= b_{k+1} - A_{k+1}x^{(k)} - A_{k+1}\Delta x^{(k)} \\ &\approx b_k - A_k x^{(k)} - A_{k+1}\Delta x^{(k)} \\ &= r_k - A_{k+1}\Delta x^{(k)} \end{aligned}$$

where it is assumed that, $r_k \approx b_{k+1} - A_{k+1}x^{(k)}$.

$$\therefore r_k - A_{k+1}\Delta x^{(k)} = r_{k+1} \tag{3.6}$$

Equation (3.6) represents a minimization problem, where the norm of the new residual vector r_{k+1} has to be minimized.

$$A_{k+1}\Delta x^{(k)} \approx r_k \tag{3.7}$$

$$\Delta x^{(k)} = A_{k+1}^{-1} r_k \tag{3.8}$$

$\Delta x^{(k)}$ should be chosen such that the norm of the new residual vector is minimum. The matrix A_{k+1}^{-1} can be expressed as a polynomial of degree $N-1$ in A_{k+1} using Cayley-Hamilton therom.

Proof:

Consider a matrix A of size $N \times N$. Its characteristic equation can be written as

$$\mid \lambda I - A \mid = \lambda^N + a_1\lambda^{N-1} + a_2\lambda^{N-2} + \ldots + a_{N-1}\lambda + a_N = 0$$

where λ is an eigenvalue of A and I is an identity matrix of size $N \times N$.

The Cayley-Hamilton theorem states that the matrix A satisfies its own characteristic equation.

$$\begin{aligned} \therefore \quad 0 &= A^N + a_1A^{N-1} + \ldots + a_{N-1}A + a_NI \\ I &= -\frac{1}{a_N}(A^N + a_1A^{N-1} + a_2A^{N-2} + \ldots + a_{N-1}A) \\ A^{-1} &= -\frac{1}{a_N}(A^{N-1} + a_1A^{N-2} + \ldots + a_{N-1}I) \end{aligned}$$

$\therefore A^{-1}$ can be expressed as a polynomial of degree $N-1$ in A.

From the above proof, it follows that $A_{k+1}^{-1}r_k$ can be expressed as a linear combination of column vectors $A_{k+1}^j r_k$, where $j = 0, 1, 2 \ldots, N-1$. These column vectors are arranged in a matrix to form the Krylov matrix K_N of order $N \times N$. If r_k is proportional to an eigen vector of A_{k+1}, then K_N has only one independent column, since all the vectors, $r_k, A_{k+1}r_k, A_{k+1}^2r_k, \ldots, A_{k+1}^{N-1}r_k$, are linearly related. If r_k is close to an eigenvector of A_{k+1}, then the columns of K_N are linearly related, only the first few columns are essential in the solution. This is of course only an approximation but it leads to computational efficiency. Thus if only the first $m << N$ columns are used, the Krylov matrix becomes $K_m = [r_k, A_{k+1}r_k, A_{k+1}^2r_k, ..., A_{k+1}^{m-1}r_k]$ of order $N \times m$. If

the rank of K_m is m, the columns of K_m spans a subspace of order m. Now $\Delta x^{(k)}$ can be expressed as a linear combination of columns of K_m.

$$\Delta x^{(k)} = K_m y \tag{3.9}$$

where y is a column vector of size $m \times 1$. Since columns of K_m does not span the full space $\mathbb{R}^N$, (3.9) can be only approximately satisfied. This is based on the assumption that $\Delta x^{(k)}$ is confined to the Krylov subspace spanned by the columns of K_m.
Substitution of $\Delta x^{(k)}$ from (3.9) into (3.6) yields the minimization problem

$$r_k - A_{k+1} K_m y = r_{k+1} \tag{3.10}$$

where norm of r_{k+1} has to be minimized. Arnoldi's method can be used to orthogonalize the Krylov matrix K_m to get an orthonormal matrix V. Hence $\Delta x^{(k)}$ can be expressed as a linear combination of columns of V. Then Equations (3.9) and (3.10) take the form

$$\Delta x^{(k)} = V z \tag{3.11}$$

$$r_k - A_{k+1} V z = r_{k+1} \tag{3.12}$$

where z is a column vector of size $m \times 1$. The norm of r_{k+1} has to be minimized. Least Squares Solution of (3.12) is given below. Choose

$$v_1 = r_k / \|r_k\| \tag{3.13}$$

as the first vector in V. The other vectors are obtained using Arnoldi's algorithm.
For $j = 1, 2 \ldots, m$ Do:
$h_{ij} = \langle A_{k+1} v_j, v_i \rangle$ for $i = 1, 2, \ldots, j$
$w_j = A_{k+1} v_j - \sum_{i=1}^{j} h_{ij} v_i$
$h_{j+1,j} = \|w_j\|$
$v_{j+1} = w_j / h_{j+1,j}$
End Do
The above algorithm implies

$$A_{k+1} v_j = \sum_{i=1}^{j+1} h_{i,j} v_i \text{ for } j = 1, 2, \ldots, m \tag{3.14}$$

In matrix form,

$$A_{k+1}V = V_{m+1}H \tag{3.15}$$

where V_{m+1} is the matrix V augmented by the new column v_{m+1}, and H is an upper Hessenberg matrix of dimension $(m+1) \times m$ with elements h_{ij}. Substituting (3.15) in (3.12) gives

$$r_k - V_{m+1}Hz = r_{k+1} \tag{3.16}$$

To minimize the new residual it must be orthogonal to the span of $V_{m+1}H$.

Proof:

Let the linear system of equations to be solved be

$$Ax = b \tag{3.17}$$

Let x_k be the solution at k^{th} iteration and r_k be the corresponding residual vector.

$$r_k = b - Ax_k \tag{3.18}$$

Let $x = x_k + z$ be any vector in $x_k + K$, where $z \in K$ and K is the Krylov subspace spanned by the vectors $r_k, Ar_k, A^2r_k, ..., A^{m-1}r_k$. Let $\overline{x}$ be the approximate solution, such that $\overline{x} = x_k + \overline{z}$, where $\overline{x} \in x_k + K$ and $\overline{z} \in K$.

The approximate solution for equation (3.17) is chosen from $x_k + K$. The condition to be satisfied to get the minimum possible residual norm can be derived as follows.

$$\begin{aligned}
\|b - Ax\|^2 &= \|b - A[(x - \overline{x}) + \overline{x}]\|^2 \\
&= \|b - A\overline{x} - A(x - \overline{x})\|^2 \\
&= \langle b - A\overline{x} - A(x - \overline{x}) \ , \ b - A\overline{x} - A(x - \overline{x})\rangle \\
&= \langle b - A\overline{x}, b - A\overline{x}\rangle - 2\langle b - A\overline{x}, A(x - \overline{x})\rangle + \langle A(x - \overline{x}), A(x - \overline{x})\rangle \\
&= \|b - A\overline{x}\|^2 - 2\langle b - A\overline{x}, A(x - \overline{x})\rangle + \|A(x - \overline{x})\|^2
\end{aligned}$$

$$\|b - Ax\|^2 = \|b - A\overline{x}\|^2 - 2\langle b - A\overline{x}, A(x - \overline{x})\rangle + \|A(x - \overline{x})\|^2 \tag{3.19}$$

$$\overline{x} = x_k + \overline{z} \text{ and } x = x_k + z$$

$$\therefore x - \overline{x} = z - \overline{z}$$

Since both z and $\overline{z}$ are elements of K,

$z - \overline{z} \in K$

$\therefore\ x - \overline{x} \in K$

and $A(x - \overline{x}) \in AK$

If

$$\langle b - A\overline{x}, A(x - \overline{x})\rangle = 0, \tag{3.20}$$

then (3.19) becomes

$$\|b - Ax\|^2 = \|b - A\overline{x}\|^2 + \|A(x - \overline{x})\|^2 \tag{3.21}$$

$$\therefore \|b - A\overline{x}\| \leq \|b - Ax\| \tag{3.22}$$

$\therefore\ \overline{x}$ gives the minimum possible residual norm. It means, to get the minimum residual norm, $\overline{x}$ should be chosen such that (3.20) is satisfied.

$$\Rightarrow b - A\overline{x} \perp AK$$

$$\text{If } \ b - A\overline{x} = r_{k+1},\ r_{k+1} \perp AK$$

Since K is orthonormalized to get V by a modified Gram-Schmidt process in the Arnoldi's algorithm, $r_{k+1} \perp A_{k+1}V$

From Arnoldi's algorithm $A_{k+1}V = V_{m+1}H \therefore r_{k+1} \perp V_{m+1}H$

Premultiplying (3.16) by $(V_{m+1}H)^T$ and using $(V_{m+1}H)^T r_{k+1} = 0$ by the above proof gives,

$$H^T H z = H^T {V_{m+1}}^T r_k \tag{3.23}$$

Equation (3.23) is equivalent to the least squares problem

$$Hz = {V_{m+1}}^T r_k \tag{3.24}$$

$$Hz = b \tag{3.25}$$

where $b = [\beta, 0, 0, \ldots, 0]^T$ is an $(m+1) \times 1$ vector and $\beta = \|r_k\|$. Solving this least squares problem gives z. Substitution of this z in (3.11) gives $\Delta x^{(k)}$. Subsequently (3.5) yields $x^{(k+1)}$ and from (3.2) r_{k+1} is obtained.

3.2 The Power Flow Problem

For an n bus system if bus 1 is the slack bus, buses $2, 3, \ldots, g$ are the PV buses and buses $g+1, \ldots, n$ are PQ buses then the power flow equations are

$$\Delta P_i = P_i^{sp} - \sum_{k=1}^{n} V_i V_k Y_{ik} \cos(\theta_{ik} - \delta_i + \delta_k) = 0 \text{ for } i\text{=2,3,}\ldots\text{,}n \quad (3.26)$$

$$\Delta Q_i = Q_i^{sp} + \sum_{k=1}^{n} V_i V_k Y_{ik} \sin(\theta_{ik} - \delta_i + \delta_k) = 0 \text{ for } i\text{=g+1,}\ldots\text{,}n \quad (3.27)$$

3.3 Implementation Alternatives

3.3.1 Constant Matrix Alternative (least squares solution)

In this method matrix A is replaced by the Jacobian matrix used in Fast Decoupled Load Flow. Solving the least squares problem (3.24) gives z.

3.3.2 Simplified Solution

Since H is an upper Hessenberg matrix the last row contains only one nonzero element. If the last element of H is set equal to zero then it becomes a regular $m \times m$ Hessenberg matrix and then V_{m+1} becomes V and b will have length m. Then (3.24) becomes a set of linear equations.

3.3.3 The Quasi-Newton Alternative

In Arnoldi's algorithm A is not needed explicitly since it is used only as an operator to obtain the vector Av_j. This can be calculated directly as the directional derivative of $f(x)$.

$$f(x) = p_{sp} - p(x) \quad (3.28)$$

where p_{sp} represents specified active and reactive powers and p represents the calculated active and reactive powers. Thus linearizing $p(x)$ at $x_k + \epsilon v_j$, around x_k gives

$$p(x_k + \epsilon v_j) = p(x_k) + J_k \epsilon v_j \quad (3.29)$$

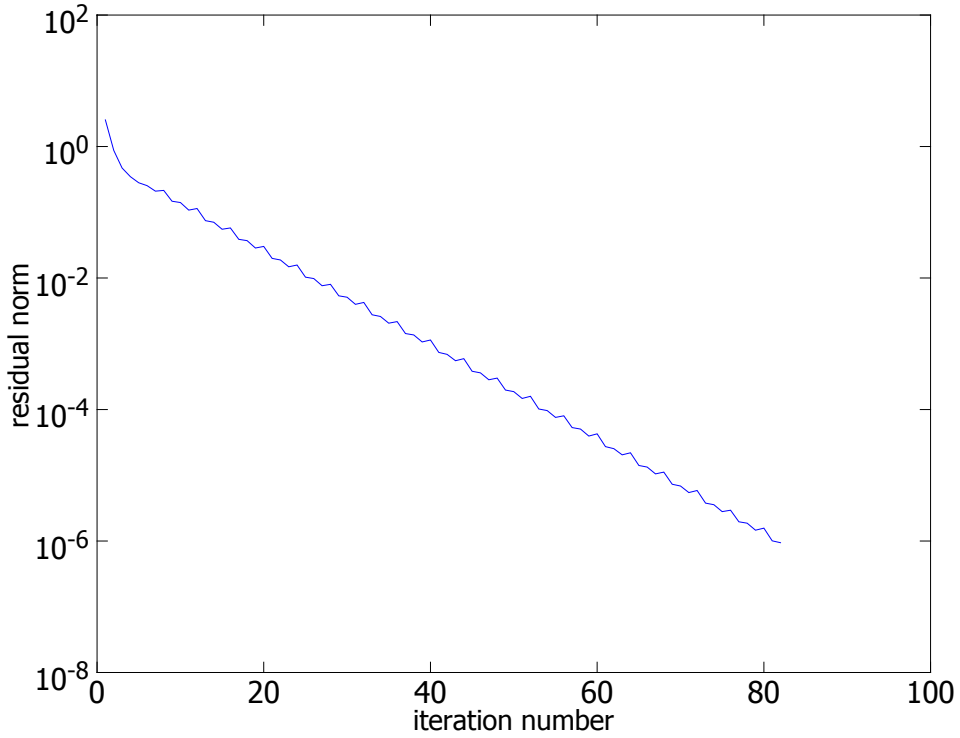

Figure 3.1: Constant matrix alternative (least squares solution) for m=12 for 118 bus system

This implies, with Jacobian J_k replaced by A,

$$Av_j = \frac{p(x_k + \epsilon v_j) - p(x_k)}{\epsilon} \tag{3.30}$$

3.4 Numerical Results

3.4.1 Results for IEEE 118 bus system

The number of iterations taken for the residual norm to be less than 10^{-6} and corresponding time taken for the execution of the program for different values of m for constant matrix alternative (least squares solution) are given in Table 3.1.

Figures 3.1 and 3.2 show the variation of residual norm (Euclidean norm of the residual vector) with iteration number for $m = 12$ and 28 respectively. It can be seen that as the value of m increases from 10 to 28 the number of iterations required for convergence decreases but further increase in the value of m results in an increase in the time required for the execution of the program.

Table 3.2 shows the number of iterations taken for the residual norm to be less than

Table 3.1: Constant matrix method (least squares solution) for 118 bus system

Value of m	No.of iterations required for the residue to be less than 10^{-6}	Time taken for the execution of the program (in seconds)
10	137	133.51
12	82	80.699
14	55	54.756
15	43	43.056
16	40	40.529
17	35	35.599
18	28	28.876
19	27	28.002
20	25	26.099
21	24	24.835
22	23	24.024
23	22	23.182
24	21	22.308
25	21	22.370
26	19	20.592
27	19	20.514
28	17	18.502
29	17	18.595
30	17	18.767
31	17	18.720
32	17	18.736
33	17	18.798
34	17	18.923
35	17	18.970
36	17	19.001
37	17	19.048
38	17	19.204
39	21	23.462
40	28	31.060
41	30	33.275

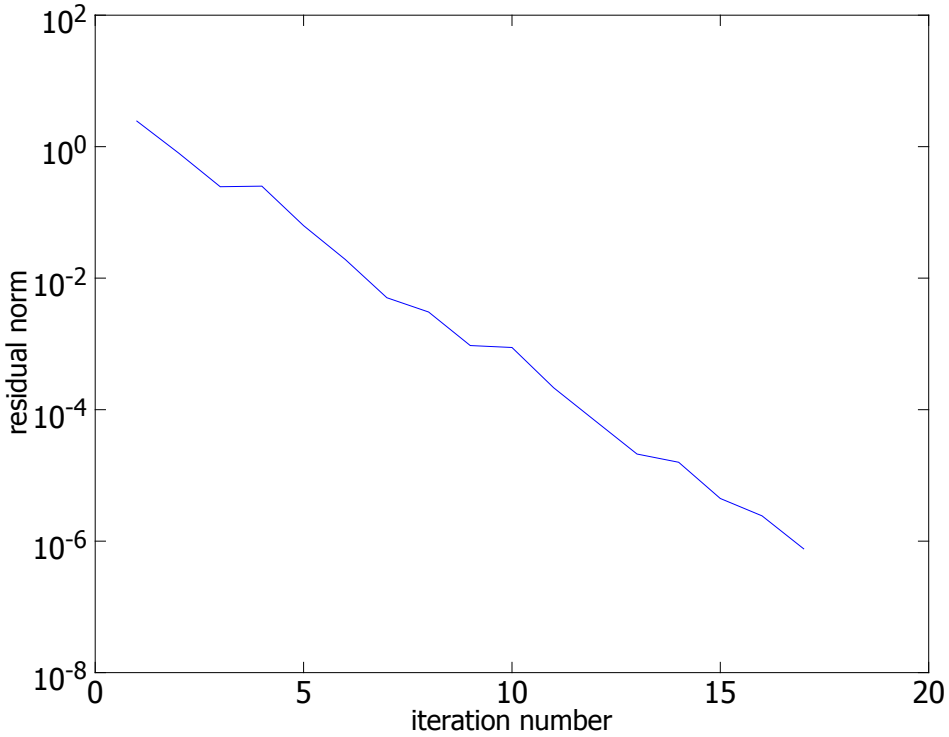

Figure 3.2: Constant matrix alternative (least squares solution) for m=28 for 118 bus system

10^{-6} for different values of m and the corresponding time taken for the execution of the program for the constant matrix simplified approach. Figures 3.3 and 3.4 show the variation of norm of the residue with iteration number for m=15 and 21 respectively. In this approach also, the number of iterations required for convergence decreases with increase in the value of m, till $m = 21$. Further increase in m results in an increase in the number of iterations and time required for execution. Table 3.3 shows the number of iterations for the norm of residue to be less than 10^{-6} and time taken for the execution of the program for Quasi-Newton method. Figures 3.5 and 3.6 depict the convergence pattern for Quasi-Newton method for m=45 and 48 respectively.

The results show that in Quassi-Newton alternative, convergence is obtained in less number of iterations compared to the other two methods. Since it is based on a variable A matrix, the time required per iteration is more.

The conventional Newton-Raphson power flow method takes only 13.12 seconds for the residual norm to be less than 10^{-12}, whereas the time required for all the three alternatives of KSPF are relatively more.

Table 3.2: Constant matrix simplified approach for 118 bus system

Value of m	No.of iterations required for the residue to be less than 10^{-6}	Time taken for the execution of the program(in seconds)
10	135	134.25
12	71	69.826
13	55	54.803
14	47	46.831
15	41	41.215
16	39	39.406
17	32	32.542
18	28	28.891
19	24	24.960
20	22	22.994
21	18	19.157
22	21	22.183
23	19	20.186
24	21	22.292
25	20	21.372
26	18	19.422
27	20	21.512
28	18	19.406
29	18	19.547
30	18	19.578
31	20	21.762
32	22	23.915
33	21	23.026
34	27	29.385
35	29	31.606

Table 3.3: Quasi-Newton alternative for 118 bus system

Value of m	No.of iterations required for the norm of residue to be less than 10^{-6}	Time taken for the execution of the program(in seconds)
20	35	694.53
25	22	539.40
35	12	410.25
40	9	349.74
45	8	348.16
48	7	325.48
50	7	339.25

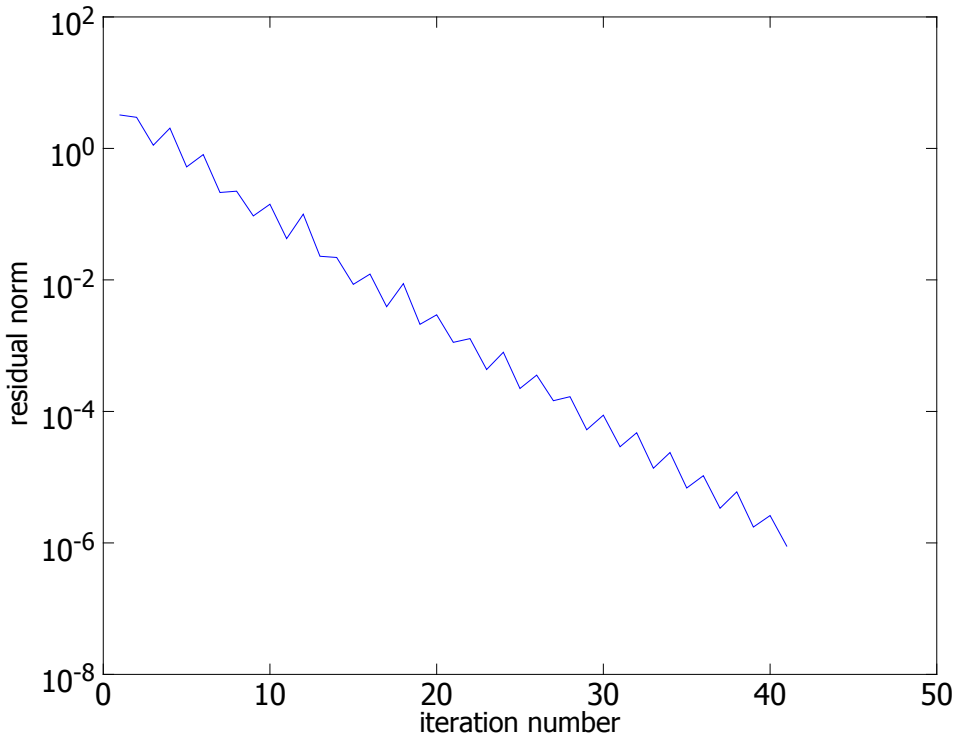

Figure 3.3: Constant matrix simplified approach for m=15 for 118 bus system

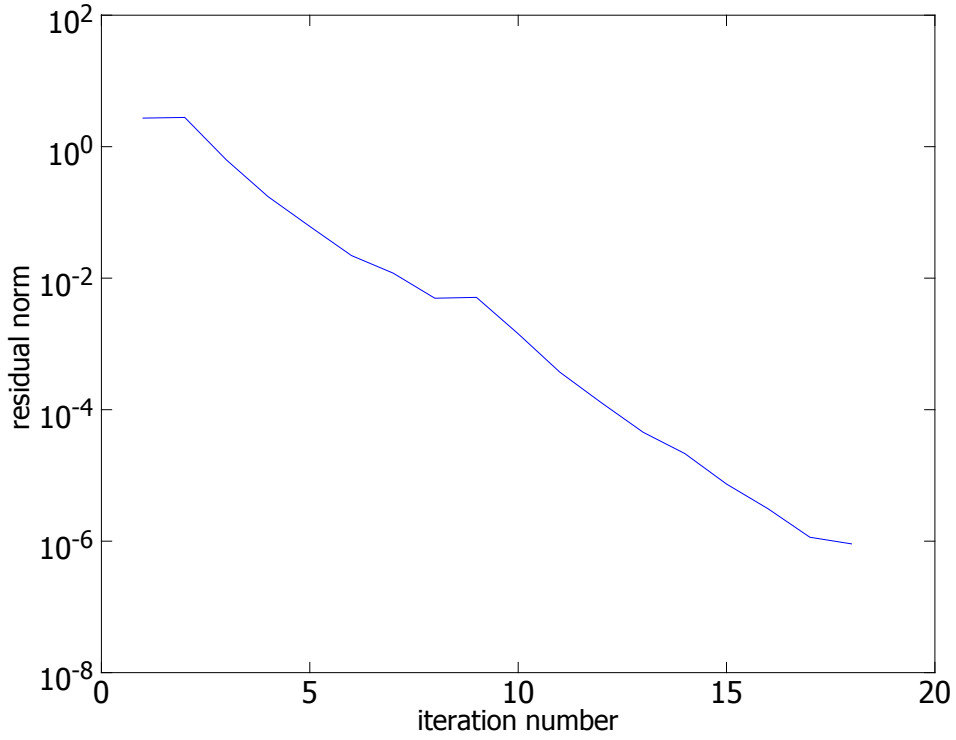

Figure 3.4: Constant matrix simplified approach for m=21 for 118 bus system

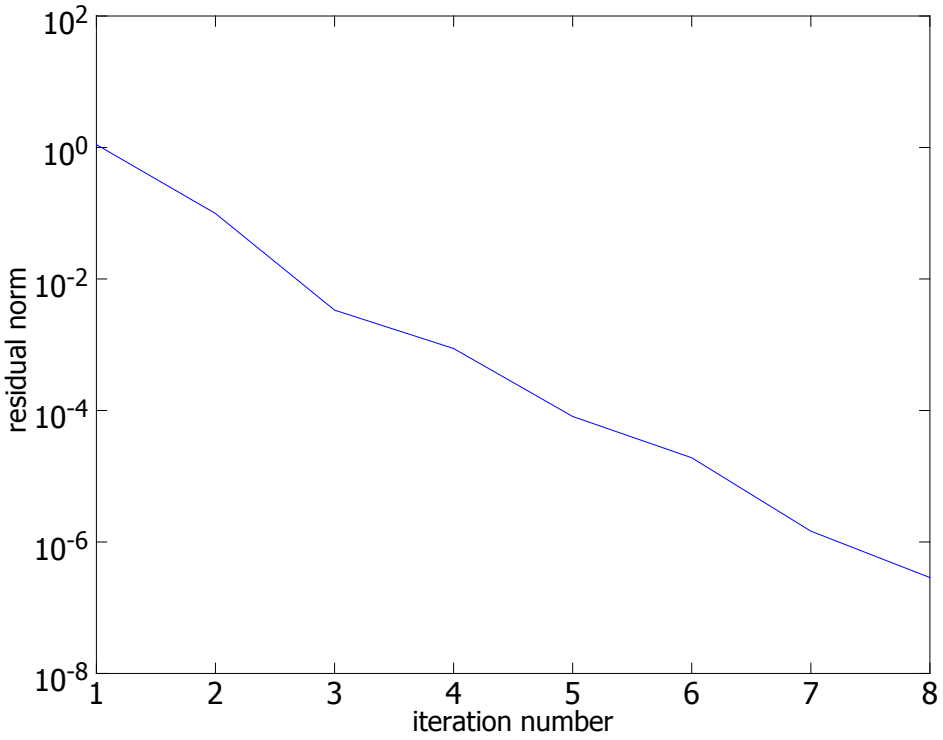

Figure 3.5: Quasi-Newton alternative for m=45 and $\epsilon = 0.001$ for 118 bus system

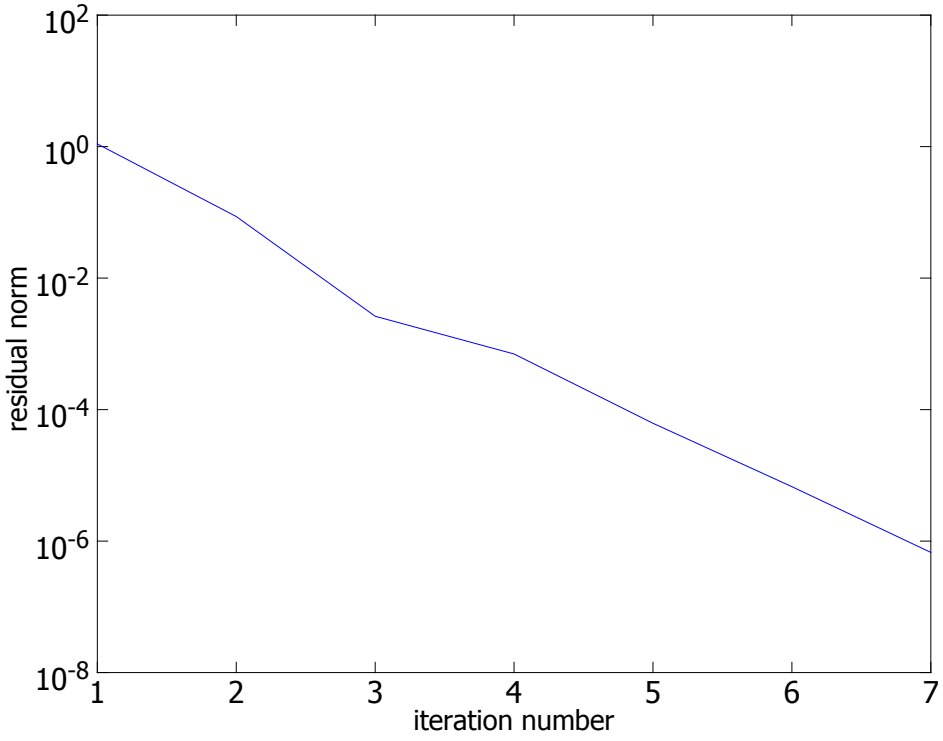

Figure 3.6: Quasi-Newton alternative for m=48 and $\epsilon = 0.001$ for 118 bus system

3.4.2 Results for IEEE 300 bus system

Table 3.4: Constant matrix method (least squares solution) for 300 bus system

Value of m	Number of iterations performed	Norm of residue at the end of iterations	Time taken for the execution of the program(in seconds)
23	31	4.6517	232.75
25	31	0.60052	232.77
26	31	0.55901	233.11

Table 3.5: Quasi-Newton implementation for 300 bus system

Value of m	Number of iterations performed	Norm of residue at the end of iterations	Time taken for the execution of the program
8	5	4.8326	321.44
25	3	1.112	786.02
40	3	0.85899	895.33

Constant matrix implementation alternatives failed to converge in 300 bus test system. Quasi-Newton alternative was slow and norm of the residue was almost constant after initial iterations. Results are given in table 3.4 and 3.5. The conventional Newton-Raphson power flow method takes 1.96 minutes for the norm of the residue to be less than 10^{-6}.

CHAPTER 4

Selective Eigenvalue Computation Techniques

4.1 Newton's Method

The linearized power system model for small signal stability analysis can be represented as

$$\begin{bmatrix} \dot{x} \\ 0 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ V \end{bmatrix} \tag{4.1}$$

The state vector x can be partitioned as $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, where x_1 is of size $m_1 \times 1$ and x_2 is of size $m_2 \times 1$.

x_1 will be the retained vector and the eigenvalues associated with x_1 will be of interest and x_2 can be eliminated. The partitioned form of (4.1) will be

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ 0 \end{bmatrix} = \begin{bmatrix} A_{g1} & A_{12} & B_1 \\ A_{21} & A_{g2} & B_2 \\ C_1 & C_2 & D_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ V \end{bmatrix} \tag{4.2}$$

Taking Lapalace Transform of (4.2) gives

$$\begin{bmatrix} sX_1(s) \\ sX_2(s) \\ 0 \end{bmatrix} = \begin{bmatrix} A_{g1} & A_{12} & B_1 \\ A_{21} & A_{g2} & B_2 \\ C_1 & C_2 & D_1 \end{bmatrix} \begin{bmatrix} X_1(s) \\ X_2(s) \\ V \end{bmatrix} \tag{4.3}$$

The system of equations (4.3) can be rewritten as

$$\begin{bmatrix} sX_1(s) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} A_{g1} & A_{12} & B_1 \\ A_{21} & A_{g2} - sI & B_2 \\ C_1 & C_2 & D_1 \end{bmatrix} \begin{bmatrix} X_1(s) \\ X_2(s) \\ V(s) \end{bmatrix} \tag{4.4}$$

$$\Rightarrow sX_1 = (A_{g1} - BD^{-1}(s)C)X_1(s) \tag{4.5}$$

where,

$$\begin{aligned} B &= \begin{bmatrix} A_{12} & B_1 \end{bmatrix} \\ C &= \begin{bmatrix} A_{21} \\ C_1 \end{bmatrix} \\ D(s) &= \begin{bmatrix} A_{g2} - sI & B_2 \\ C_2 & D_1 \end{bmatrix} \end{aligned}$$

Let,

$$G(s) = D^{-1}(s)C \tag{4.6}$$

$$\therefore D(s)G(s) = C \tag{4.7}$$

Differentiating the above expression gives

$$\frac{dD(s)}{ds}G(s) + D(s)\frac{dG(s)}{ds} = 0 \tag{4.8}$$

Since

$$\begin{aligned} D(s) &= \begin{bmatrix} A_{g2} - sI & B_2 \\ C_2 & D_1 \end{bmatrix} \\ \frac{dD(s)}{ds} &= \begin{bmatrix} -I & 0 \\ 0 & 0 \end{bmatrix} \triangleq -I_0 \end{aligned}$$

(4.8) becomes

$$-I_0G(s) + D(s)\frac{dG(s)}{ds} = 0 \tag{4.9}$$

$$\therefore \frac{dG(s)}{ds} = D^{-1}(s)I_0G(s) \tag{4.10}$$

Expanding $G(s)$ as a Taylor series,

$$G(s) = G(s_k) + \left.\frac{dG(s)}{ds}\right|_{s=s_k} + \text{ higher order terms} \tag{4.11}$$

Substituting for $G(s)$, $G(s_k)$ and $\frac{dG(s)}{ds}$ and neglecting higher order terms lead to

$$G(s) = D^{-1}(s)C = D^{-1}(s_k)C + D^{-1}(s_k)I_0D^{-1}(s_k)C(s - s_k) \tag{4.12}$$

(4.5) becomes,

$$\begin{aligned} sX_1 &= \Big(A_{g1} - BD^{-1}(s_k)C - BD^{-1}(s_k)I_0D_{-1}(s_k)C(s - s_k)\Big)X_1 \\ &= \Big(A_{g1} - BD^{1}(s_k)C + BD^{-1}(s_k)I_0D^{-1}(s_k)Cs_k\Big)X_1 - sBD^{-1}(s_k)I_0D^{-1}(s_k)CX_1 \end{aligned}$$

$$s\Big(I + BD^{-1}(s_k)I_0D^{-1}(s_k)C\Big)X_1 = \Big(A_{g1} - BD^{-1}(s_k)C + BD^{-1}(s_k)I_0D^{-1}(s_k)Cs_k\Big)X_1 \tag{4.13}$$

Define,

$I + BD^{-1}(s_k)I_0D^{-1}(s_k)C = J(s_k)$

(4.13) becomes

$$J(s_k)sX_1 = \Big(A_{g1} - BD^{-1}(s_k)C + (J(s_k) - I)s_k\Big)X_1 \tag{4.14}$$

$$sX_1 = J(s_k)^{-1}\Big(A_{g1} - BD^{-1}(s_k)C + (J(s_k) - I)s_k)\Big)X_1 \tag{4.15}$$

The improved guesses for eigenvalues associated with the vector x_1 are obtained by iterative solution of the equation given by

$$\Big|sI - J(s_k)^{-1}[A_{g1} - BD^{-1}(s_k)C + (J(s_k) - I)s_k)]X_1\Big| = 0 \tag{4.16}$$

If x_1 is a scalar i.e., $m_1 = 1$

$$s_{k+1} = J(s_k)^{-1}\Big(A_{g1} - BD^{-1}(s_k)C + (J(s_k) - 1)s_k\Big)X_1 \tag{4.17}$$

The iteration proceeds till s_k converges to one of the eigenvalues.

4.2 AESOPS Algorithm

In AESOPS algorithm (Analysis of Essentially Spontaneous Oscillations in Power Systems), the system matrix is organized as follows, retaining only $\Delta\omega_i(s)$.

$$\begin{bmatrix} s\Delta\omega_i(s) \\ s\Delta\delta_i(s) \\ sX_2(s) \\ 0 \end{bmatrix} = \begin{bmatrix} a_{i1} & a_{i2} & B_i & \\ 1 & 0 & 0 & 0 \\ C_{11} & C_{12} & A_{22} & B_2 \\ & & C_2 & D_1 \end{bmatrix} \begin{bmatrix} \Delta\omega_i(s) \\ \Delta\delta_i(s) \\ \Delta X_2(s) \\ \Delta Y_(s) \end{bmatrix} \tag{4.18}$$

Where B_i is a row vector and C_{11} ans C_{12} are column vectors, and

$$a_{i1} = -\frac{D_i\omega_s}{2H_i},\ a_{i2} = -\frac{\omega_s E_i V_i}{2H_i X'_{d_i}} cos(\delta_i - \phi_i)$$

Since

$$\Delta\delta_i(s) = \frac{\Delta\omega_i(s)}{s}$$

(4.18) becomes

$$s\Delta\omega_i(s) = \left(a_{i1} + \frac{a_{i2}}{s}\right)\Delta\omega_i(s) + B_i \begin{bmatrix} \Delta X_2(s) \\ \Delta Y(s) \end{bmatrix} \tag{4.19}$$

$$0 = \left(C_{11} + \frac{C_{12}}{s}\right)\Delta\omega_i(s) + \begin{bmatrix} A_{22} - sI & B_2 \\ C_2 & D_1 \end{bmatrix} \begin{bmatrix} \Delta X_2(s) \\ \Delta Y(s) \end{bmatrix} \tag{4.20}$$

Eliminating $\Delta X_2(s)$ and $\Delta Y(s)$ between these two equations gives

$$s\Delta\omega_i(s) = \left(a_{i1} + \frac{a_{i2}}{s} - B_i D^{-1}(s)\left(C_{11} + \frac{C_{12}}{s}\right)\right)\Delta\omega_i(s) \tag{4.21}$$

where,

$$D(s) = \begin{bmatrix} A_{22} - sI & B_2 \\ C_2 & D_1 \end{bmatrix}$$

The eigenvalue of this problem is obtained by solving the scalar equation

$$s + \left(-a_{i1} - \frac{a_{i2}}{s} + B_i D^{-1}(s)\left(C_{11} + \frac{C_{12}}{s}\right)\right) = 0 \tag{4.22}$$

Let

$$J(s) = s + \left(-a_{i1} - \frac{a_{i2}}{s} + B_i D^{-1}(s)\left(C_{11} + \frac{C_{12}}{s}\right)\right)$$

Using Newton-Raphson method for the iterative solution of a nonlinear equation,

$$s_{k+1} = s_k - \frac{J(s_k)}{J'(s_k)} \tag{4.23}$$

From equation (4.6)

$$G(s) = D^{-1}(s)C$$
$$\frac{dG(s)}{ds} = \frac{d}{ds}(D^{-1}(s))C$$

Using (4.10),

$$D^{-1}(s)I_0G(s) = \frac{d}{ds}(D^{-1}(s))C \tag{4.24}$$
$$\therefore \quad D^{-1}(s)I_0D^{-1}(s)C = \frac{d}{ds}(D^{-1}(s))C \tag{4.25}$$

$$\therefore \frac{d}{ds}(D^{-1}(s)) = D^{-1}(s)I_0D^{-1}(s) \tag{4.26}$$

Using (4.26) the Jacobian $J'(s_k)$ of $J(s_k)$ is given by

$$\begin{aligned} J'(s_k) =& \left.\frac{\partial J(s)}{\partial s}\right|_{s=s_k} \\ =& 1 + \frac{a_{i2}}{s_k^2} - B_i D^{-1}(s_k)\frac{C_{12}}{s_k^2} + B_i D^{-1}(s_k) I_0 D^{-1}(s_k)\left(C_{11} + \frac{C_{12}}{s_k}\right) \end{aligned} \tag{4.27}$$

$J'(s_k)$ can be approximated as

$$J'(s_k) \approx 1 + \frac{a_{i2}}{s_k^2} - B_i D^{-1}(s_k)\frac{C_{12}}{s_k^2} \tag{4.28}$$

4.3 Numerical Results

The western regional grid of Indian power system consisting of 493 buses and 193 generators is used for eigenvalue analysis. Generators are represented using classical model. So that δ and ω are the state variables and the total number of state variables

were 386. Damping coefficient is assumed to be zero for all generators.

The QR method computes all the 386 eigenvalues in 0.4386 seconds. The results obtained on 493 bus system on applying Newton's method and AESOPS algorithm are tabulated below.

The results obtained on applying Newton's method, to find out an eigenvalue associated with the rotor angle of the first generator for different initial values, are given in Table 4.1. It can be seen that the number of iterations required for convergence depends on the initial guess. Based on the initial guess, Newton's method converges to a particular eigenvalue, among the different eigenvalues associated with the state variable under coonsideration.

Table 4.1: Newton's Method for $i = 1$

s_0	Number of iterations	Eigenvalue(s)	Time taken (in seconds)
$j\pi$	13	$j8.496246$	67.44
$j1.6\pi$	9	$j8.496246$	47.03
$j1.8\pi$	8	$j8.496246$	41.81
$j2\pi$	7	$j6.290521$	36.53
$j2.2\pi$	5	$j7.817677$	26.09
$j2.4\pi$	6	$j7.537722$	31.31
$j2.6\pi$	8	$j8.814284$	41.79
$j2.8\pi$	5	$j8.814284$	26.07
$j3\pi$	6	$j9.225411$	31.37
$j3.2\pi$	5	$j9.225411$	26.18
$j3.4\pi$	6	$j9.225411$	31.26
$j3.6\pi$	6	$j9.225411$	31.29
$j3.8\pi$	6	$j9.225411$	31.42
$j4\pi$	5	$j9.950887$	26.08
$j5\pi$	6	$j9.225411$	31.18
$j5.6\pi$	7	$j9.225411$	36.33
$j6\pi$	6	$j9.225411$	31.34

The AESOPS program computes an elecromechanical oscillation mode of interest based on a good initial guess. The results obtained on using the true Jacobian (equation (4.27)) and a nearly true Jacobian (equation (4.28)) for three different states are tabulated below.

Table 4.2: AESOPS algorithm with true Jacobian for $i = 1$

s_0	Number of iterations	Eigenvalue(s)	Time taken (in seconds)
$j\pi$	15	$j9.225412$	91.49
$j2\pi$	16	$j9.225412$	97.73
$j2.8\pi$	15	$j9.225412$	91.43
$j3\pi$	13	$j9.225412$	79.29
$j3.6\pi$	14	$j9.225412$	85.24
$j3.8\pi$	14	$j9.225412$	84.59
$j4\pi$	15	$j9.225412$	90.79

Table 4.3: AESOPS algorithm with true Jacobian for $i = 26$

s_0	Number of iterations	Eigenvalue(s)	Time taken (in seconds)
$j\pi$	22	$j11.00080$	134.7
$j2\pi$	23	$j11.00080$	140.2
$j2.4\pi$	20	$j11.00080$	121.9
$j3\pi$	22	$j11.00080$	134.1
$j3.2\pi$	19	$j11.00080$	114.72
$j3.4\pi$	17	$j11.00080$	103.44
$j3.6\pi$	20	$j11.00080$	120.94

Table 4.4: AESOPS algorithm with true Jacobian for $i = 64$

s_0	Number of iterations	Eigenvalue(s)	Time taken (in seconds)
$j\pi$	25	$j10.966854$	152.08
$j2\pi$	25	$j10.966854$	151.17
$j3\pi$	24	$j10.966854$	145.15
$j3.2\pi$	23	$j10.966854$	139.85
$j3.4\pi$	21	$j10.966854$	127.85
$j3.6\pi$	20	$j10.966854$	120.86
$j3.8\pi$	21	$j10.966854$	126.95

The number of iterations and the time required for convergence for both true Jacobian and nearly true Jacobian implementations decreases as the initial guess moves

Table 4.5: AESOPS algorithm with nearly true Jacobian for $i = 1$

s_0	Number of iterations	Eigenvalue(s)	Time taken (in seconds)
$j\pi$	15	$j9.225411$	27.83
$j2\pi$	16	$j9.225411$	29.75
$j2.8\pi$	14	$j9.225411$	26.18
$j3\pi$	10	$j9.225411$	18.72
$j3.6\pi$	12	$j9.225411$	22.47
$j3.8\pi$	17	$j9.225411$	31.81

Table 4.6: AESOPS algorithm with nearly true Jacobian for $i = 26$

s_0	Number of iterations	Eigenvalue(s)	Time taken (in seconds)
$j\pi$	23	$j11.00080$	43.00
$j2\pi$	23	$j11.00080$	43.02
$j2.4\pi$	20	$j11.00080$	37.43
$j3\pi$	17	$j11.00080$	31.78
$j3.2\pi$	18	$j11.00080$	33.46
$j3.4\pi$	20	$j11.00080$	37.42
$j3.6\pi$	22	$j11.00080$	41.14

Table 4.7: AESOPS algorithm with nearly true Jacobian for $i = 64$

s_0	Number of iterations	Eigenvalue(s)	Time taken (in seconds)
$j\pi$	40	$j10.966855$	74.86
$j2\pi$	25	$j10.966855$	46.70
$j3\pi$	24	$j10.966855$	44.82
$j3.2\pi$	22	$j10.966855$	41.16
$j3.4\pi$	21	$j10.966855$	39.02
$j3.6\pi$	20	$j10.966855$	37.44
$j3.8\pi$	21	$j10.966855$	39.28

close to the desired eigenvalue. The nearly true Jacobian alternative is faster compared to the true Jaciobian alternative because of the reduction in computation.

CHAPTER 5

Computation of Eigenvalues Using Arnoldi's Algorithm

Let A be an $n \times n$ matrix whose eigenvalues are to be determined. Choose a vector v_1 of norm 1. Use of A and v_1 in $m-$step Arnoldi's algorithm results in

$$\begin{aligned} AV_m &= V_{m+1}H \quad (5.1) \\ &= V_m H_m + h_{m+1,m} v_{m+1} e_m^T \quad (5.2) \end{aligned}$$

where,

V_{m+1} is an $n \times (m+1)$ matrix with $v_1, v_2, ...v_{m+1}$ as its columns.

V_m is a matrix of size $n \times m$ obtained from V_{m+1} by deleting the last column of V_{m+1}.

H is an upper Hessenberg matrix of size $(m+1) \times m$.

H_m is an $m \times m$ matrix obtained from H by deleting the last row of H.

e_m is the m^{th}column of $m \times m$ identity matrix.

$h_{m+1,m}$ is the $(m+1)^{th}$ row, m^{th} column element of H.

Since columns of V_m form an orthonormal basis,

$$V_m^T V_m = I \quad (5.3)$$

$$V_m^T v_{m+1} = 0 \quad (5.4)$$

Multiplying (5.2) by V_m^T and using (5.3) and (5.4) results in

$$V_m^T A V_m = H_m \quad (5.5)$$

Let $y_i^{(m)}$ be a right eigenvector of H_m associated with an eigenvalue $\lambda_i^{(m)}$.

$$\therefore \quad H_m y_i^{(m)} = \lambda_i^{(m)} y_i^{(m)} \quad (5.6)$$

Multiplying equation (5.2) by $y_i^{(m)}$ gives,

$$A V_m y_i^{(m)} = V_m H_m y_i^{(m)} + h_{m+1,m} v_{m+1} e_m^T y_i^{(m)} \quad (5.7)$$

Using equation (5.6), (5.7) becomes

$$AV_m y_i^{(m)} = \lambda_i^{(m)} V_m y_i^{(m)} + h_{m+1,m} v_{m+1} e_m^T y_i^{(m)} \tag{5.8}$$

$$\Rightarrow (A - \lambda_i^{(m)} I) V_m y_i^{(m)} = h_{m+1,m} v_{m+1} e_m^T y_i^{(m)} \tag{5.9}$$

where I is an identity matrix of size $n \times n$.

If $h_{m+1,m} v_{m+1} e_m^T y_i^{(m)} \rightarrow$ a zero vector, then $\lambda_i^{(m)}$ is an approximate eigenvalue of A and $V_m y_i^{(m)}$ is the corresponding approximate right eigenvector of A. $\lambda_i^{(m)}$ is known as Ritz value or Arnoldi eigenvalue estimate. $V_m y_i^{(m)}$ is called the corresponding Ritz vector of A.

5.1 Spectral Transformation

Arnoldi's algorithm converges to eigenvalues of largest modulus first, so to get eigenvalues close to a particular point in the complex plane, shift and invert concept is used [12]. Consider a matrix A of size $n \times n$. Let λ_i be the i^{th} eigenvalue of A and v_i be the corresponding right eigenvector.

$$Av_i = \lambda_i v_i \tag{5.10}$$

Let σ be a complex number,

$$\begin{aligned}(A - \sigma I)v_i &= Av_i - \sigma v_i \\ &= \lambda_i v_i - \sigma v_i \\ &= (\lambda_i - \sigma) v_i\end{aligned}$$

$$(A - \sigma I)^{-1} v_i = \frac{1}{\lambda_i - \sigma} v_i \tag{5.11}$$

$\therefore$ Eigenvalues of $(A - \sigma I)^{-1}$ are $\frac{1}{\lambda_1 - \sigma}, \frac{1}{\lambda_2 - \sigma}, \ldots . \frac{1}{\lambda_n - \sigma}$.

Hence eigenvalue of $(A - \sigma I)^{-1}$ with largest modulus is the one which is closest to σ. Therefore $(A - \sigma I)^{-1}$ is used instead of A in the Arnoldi's algorithm so that eigenvalues close to σ will converge first.

$\therefore$ If μ_i is the i^{th} eigenvalue of $(A - \sigma I)^{-1}$, then the corresponding eigenvalue of A is obtained as $\lambda_i = 1/\mu_i + \sigma$.

5.2 Application To Power Systems

5.2.1 Power System Modelling

The linearized power system model for small signal stability analysis can be represented as

$$\begin{bmatrix} \dot{x} \\ 0 \end{bmatrix} = \begin{bmatrix} J_A & J_B \\ J_C & J_D \end{bmatrix} \begin{bmatrix} x \\ V \end{bmatrix} \tag{5.12}$$

where x is the vector of state variables and V is the vector of algebraic variables. J_A, J_B, J_C and J_D are sparse matrices which depend on system parameters and operating point. From equation (5.12) it can be shown that,

$$\dot{x} = (J_A - J_B J_D^{-1} J_C)x \tag{5.13}$$

From equation (5.13), state matrix of the system is given by

$$A = J_A - J_B J_D^{-1} J_C \tag{5.14}$$

where $A \in \mathbb{R}^{n \times n}$ and is in general a dense matrix. Hence direct computation and use of A can be avoided in the algorithm by using only the augmented system of equations in (5.12) as discussed below.

5.2.2 Sparsity Oriented Implementation

Arnoldi's algorithm requires the computation of a matrix-vector product Av or $(A - \sigma I)^{-1}v$, if a spectral transformation is performed to get eigenvalues close to σ. The matrix-vector product $(A - \sigma I)^{-1}v$, where $A = J_A - J_B J_D^{-1} J_C$ is equal to u which is obtained by solving the following augmented system of equations.

$$\begin{bmatrix} J_A - \sigma I & J_B \\ J_C & J_D \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} = \begin{bmatrix} v \\ 0 \end{bmatrix} \tag{5.15}$$

where I is an identity matrix of size same as that of J_A .

Proof

From (5.15),

$$J_C u + J_D w = 0$$
$$\therefore w = -J_D^{-1} J_C u$$

and (5.15) also gives

$$(J_A - \sigma I)u + J_B w = v$$
$$\therefore (J_A - \sigma I - J_B J_D^{-1} J_C)u = v$$
$$(A - \sigma I)u = v \quad (\because \text{using(5.14)})$$

$$u = (A - \sigma I)^{-1} v \tag{5.16}$$

Hence u is the required matrix-vector product.

Solution of the equation (5.15) is given below.

$$(J_A - \sigma I)u + J_B w = v \tag{5.17}$$
$$\therefore \quad u = -(J_A - \sigma I)^{-1} J_B w + (J_A - \sigma I)^{-1} v \tag{5.18}$$

$$J_C u + J_D w = 0 \tag{5.19}$$

Substituting (5.18) in (5.19) gives

$$J_C (J_A - \sigma I)^{-1} J_B w + J_C (J_A - \sigma I)^{-1} v + J_D w = 0 \tag{5.20}$$

$$\therefore \quad (J_D - J_C (J_A - \sigma I)^{-1} J_B) w = -J_C (J_A - \sigma I)^{-1} v \tag{5.21}$$

Calculate,

$$J_{D_{eq}} = J_D - J_C (J_A - \sigma I)^{-1} J_B \tag{5.22}$$

Equation (5.21) becomes

$$J_{D_{eq}} w = -J_C (J_A - \sigma I)^{-1} v \tag{5.23}$$

Solve for w and substitute that in (5.18) to find out u.

5.2.3 Algorithm for Eigenvalue Computation Using Arnoldi's Method

1. Choose a value for m and fix the complex shift point σ.

2. Choose the starting vector v_1 of size $n \times 1$ and norm equal to 1, which is the first column of matrix V of size $n \times m$.

3. For $j = 1, 2, ...m$ Do:
   Calculate the matrix-vector product, $u_j = (A - \sigma I)^{-1} v_j$
   Calculate $h_{i,j} = \langle u_j, v_i \rangle$ for $i = 1, 2, ...j$
   $w_j = u_j - \sum_{i=1}^{j} h_{i,j} v_i$
   $h_{j+1,j} = \|w_j\|$
   $v_{j+1} = w_j / h_{j+1,j}$
   End Do
   Step 3 implies

$$A_t v_j = \sum_{i=1}^{j+1} h_{i,j} v_i \quad \text{for } j = 1, 2, ...m \tag{5.24}$$

   where $A_t = (A - \sigma I)^{-1}$.
   Equation (5.24) in matrix form can be written as,

$$A_t V_m = V_{m+1} H \tag{5.25}$$

   where V_{m+1} is the matrix V_m augmented by the column vector v_{m+1} and H is an upper Hessenberg matrix of size $(m + 1) \times m$ with elements $h_{i,j}$.

4. Calculate the eigenvalues of the matrix H_m of size $m \times m$ obtained from H by removing the last row.

5. Accept the eigenvalues for which the norm of the vector $h_{m+1,m} v_{m+1} e_m^T y_i^{(m)}$ is less than the tolerance level. Where $y_i^{(m)}$ is the i^{th} eigenvector of H_m.

6. If all required eigenvalues are obtained, exit; otherwise restart the Arnoldi's method with a new starting vector v_1.

5.2.4 Restarting the Arnoldi's method with a new starting vector

The starting vector is chosen as a weighted linear combination of the eigenvectors of A obtained in the previous iteration. The following expression is used to update the starting vector before each iteration [12].

$$v_1 = \alpha \sum_{i=1}^{m} \|(A - \lambda_i I)V_m y_i\| V_m y_i \tag{5.26}$$

where α is a normalizing scalar. It can be shown that (5.25) is equal to

$$v_1 = \alpha^{'} V_m P \overline{p} \tag{5.27}$$

where, $P = [\ y_1^{(m)}\ y_2^{(m)} ....y_m^{(m)}\ ]$ is an $m \times m$ matrix whose columns are right eigenvectors of H_m and $\overline{p} = [\ |y_{m1}|\ |y_{m2}| ....|y_{mm}|\ ]^T$.
Here y_{mi} is the last element of $y_i^{(m)}$ and $\alpha^{'}$ is a normalizing scalar.

Proof
By definition,

$$H_m y_i^{(m)} = \lambda_i^{(m)} y_i^{(m)} \text{ and} \tag{5.28}$$

$$e_m^T y_i^{(m)} = y_{mi} \tag{5.29}$$

where e_m is the m^{th} column of an identity matrix of size $m \times m$.

$$\begin{aligned} \|(A - \lambda_i I)V_m y_i^{(m)}\| &= \|(AV_m - \lambda_i^{(m)} V_m) y_i^{(m)}\| && (5.30)\\ &= \|(V_m H_m + h_{m+1,m} v_{m+1} e_m^T - \lambda_i^{(m)} V_m) y_i^m\| && (5.31)\\ &= \|h_{m+1,m}\| |y_{mi}| \quad (\because \text{using } (5.28) \text{ and } (5.29)) && (5.32) \end{aligned}$$

Let $\alpha^{'} = \alpha / |h_{m+1,m}|$. Substituting (5.32) in (5.26) gives

$$v_1 = \alpha^{'} \sum_{i=1}^{m} |y_{mi}| V_m y_i^{(m)} = \alpha^{'} V_m P \overline{p} \tag{5.33}$$

5.3 Numerical Results

The western regional grid of Indian power system consisting of 493 buses and 193 generators is used for eigenvalue analysis. Generators are represented using classical model. So that δ and ω are the state variables and the total number of state variables were 386. Damping coefficient is assumed to be zero for all generators.

The QR method computes all the 386 eigenvalues in 0.4386 seconds. The results obtained on implementing Arnoldi's algorithm for selective eigenvalue computation on 493 bus system are tabulated below.

Table 5.1: Arnoldi's Method for $m = 10$

σ	Number of iterations	Number of converged eigenvalues	Eigenvlaue(s)	Time taken (in seconds)
$j1.2\pi$	1	1	$j4.067029$	0.3744
$j1.6\pi$	1	1	$j5.106918$	0.3744
$j2\pi$	1	1	$j6.290521$	0.3744
$j2.4\pi$	1	1	$j7.537722$	0.3744
$j2.8\pi$	1	2	$j8.814284$ $j8.772989$	0.3900
$j3.4\pi$	1	1	$j10.689248$	0.3744
$j4\pi$	1	1	$j12.558579$	0.3744
$j4.4\pi$	1	1	$j13.826015$	0.3744
$j4.8\pi$	1	1	$j15.123944$	0.3900
$j5.4\pi$	1	1	$j17.100873$	0.3744
$j6\pi$	1	1	$j18.822872$	0.3744

Table 5.2: Arnoldi's Method for $m = 15$

σ	Number of iterations	Number of converged eigenvalues	Eigenvlaue(s)	Time taken (in seconds)
$j1.2\pi$	1	2	$j4.067029$ $j4.369100$	0.3900
$j1.6\pi$	1	3	$j4.857289$ $j5.1069185$ $j5.6907018$	0.3900
$j2\pi$	1	3	$j5.974755$ $j6.290521$ $j6.566507$	0.3900
$j2.4\pi$	1	2	$j7.431196$ $j7.537722$	0.4056
$j2.8\pi$	1	2	$j8.772989$ $j8.814284$	0.3900
$j3.2\pi$	1	2	$j9.950888$ $j9.963008$	0.4056
$j3.6\pi$	1	1	$j11.254755$	0.4056
$j4\pi$	1	2	$j12.549964$ $j12.558579$	0.4056
$j4.4\pi$	1	3	$j13.761314$ $j13.826015$ $j13.858209$	0.4056
$j4.8\pi$	1	2	$j14.939408$ $j15.123944$	0.4056
$j5.2\pi$	1	2	$j16.478479$ $j16.507418$	0.4056

Table 5.2 – Arnoldi's Method for $m = 15$ (continued)

σ	Number of iterations	Number of converged eigenvalues	Eigenvlaue(s)	Time taken (in seconds)
$j5.6\pi$	1	2	$j17.546579$ $j17.777433$	0.4056
$j6\pi$	1	3	$j18.480451$ $j18.693528$ $j18.822872$	0.4056

Table 5.3: Arnoldi's Method for $m = 20$

σ	Number of iterations	Number of converged eigenvalues	Eigenvlaue(s)	Time taken (in seconds)
$j1.2\pi$	1	3	$j4.067029$ $j4.369100$ $j4.857289$	0.4212
$j1.6\pi$	1	6	$j4.067029$ $j4.369100$ $j4.857289$ $j5.106919$ $j5.426447$ $j5.442811$	0.4212
$j2\pi$	1	4	$j5.97455$ $j6.290521$ $j6.566507$ $j6.690413$	0.4368

Table 5.3 – Arnoldi's Method for $m = 20$ (continued)

σ	Number of iterations	Number of converged eigenvalues	Eigenvlaue(s)	Time taken (in seconds)
$j2.4\pi$	1	3	$j7.240610$ $j7.431196$ $j7.537722$	0.4212
$j2.8\pi$	1	4	$j8.814284$ $j8.772989$ $j8.882649$ $j8.898034$	0.4212
$j3.2\pi$	1	5	$j9.872296$ $j9.950887$ $j9.963008$ $j10.233450$ $j10.273171$	0.4368
$j3.6\pi$	1	4	$j11.141174$ $j11.254755$ $j11.474732$ $j11.670321$	0.4212
$j4\pi$	1	3	$j12.549964$ $j12.558579$ $j12.610222$	0.4212
$j4.4\pi$	1	3	$j13.761314$ $j13.826015$ $j13.858209$	0.4212

Table 5.3 – Arnoldi's Method for $m = 20$ (continued)

σ	Number of iterations	Number of converged eigenvalues	Eigenvlaue(s)	Time taken (in seconds)
$j4.8\pi$	1	5	$j14.895063$ $j14.939408$ $j15.123944$ $j15.199654$ $j15.320045$	0.4368
$j5.2\pi$	1	3	$j16.478479$ $j16.507418$ $j17.100872$	0.4212
$j5.6\pi$	1	6	$j17.100873$ $j17.546579$ $j17.774336$ $j17.831468$ $j18.012268$	0.4212
$j6\pi$	1	7	$j18.012268$ $j18.480451$ $j18.693528$ $j18.822872$ $j19.577572$ $j19.743098$ $j20.585525$	0.4368

The Arnoldi's algorithm converges to one or more eigenvalues close to the complex shift point σ. In all the cases considered, in the first iteration itself convergence is obtained. As the value of m increases the number of eigenvalues converged icreases. For values of m less than or equal to 20, the time taken for the execution of Arnoldi's method is less than that of QR method.

5.3.1 Numerical Results for Sparsity Oriented Implementation

The numerical results obtained for a sparsity oriented implementation as discussed in section 4.2.2 with a shift point of $\sigma = j4\pi$ are given in Table 4.4. The results show that the sparsity oriented implementation takes more time compared to QR method and Arnoldi's method without sparse implementation for the 493 bus system with 386 eigenvalues.

Table 5.4: Arnoldi's Method for $\sigma = j4\pi$ with Sparsity Oriented Implementation

m	Number of iterations	Eigenvalue(s)	Time taken (in seconds)
10	1	$j12.555700$	0.62
12	1	$j12.555700$	0.80
15	1	$j12.555700$ $j12.619608$	1.15
18	1	$j12.555700$ $j12.619608$ $j12.463132$	1.62
20	1	$j12.555700$ $j12.619608$ $j12.463132$	2.00
25	1	$j12.555700$ $j12.619608$ $j12.463132$ $j12.655800$	3.07
30	1	$j12.555700$ $j12.619608$ $j12.463132$ $j12.655800$ $j12.370909$ $j12.366881$	4.34

CHAPTER 6

Conclusions

6.1 Conclusions

In this work Arnoldi's algorithm is applied to power flow and small signal stability analysis of power systems. Based on the numerical results presented in the previous chapters, the following conclusions can be made.

The first part of this work evaluated the performance of KSPF in large systems compared to conventional Newton Raphson power flow. The KSPF method takes more time for convergence compared to Newton Raphson method. This is because, in constant matrix alternative, a truncated part of approximate Jacobian is used which requires more number of iterations for convergence. In the Quassi-Newton alternative direct evaluation of the Jacobian is avoided but the number of computations are more and takes more time per iteration. Another disadvantage of KSPF is that convergence depends on the eigenspectrum of the Jacobian and hence its success cannot be assured.

The second part of this work evaluated the performance of Arnoldi's method in selectively computing the eigenvalues of a larger system. The results show that Arnoldi's method gives the desired eigenvalues in less time compared to QR method which evaluates all the eigenvalues. Arnoldi's method is found to be less time consuming compared to other selective computation techniques in the literature such as Newton's method and AESOPS method. One more advantage of Arnoldi's method is that it can compute eigenvalues using the augmented system of equations and hence sparsity oriented implementation is possible unlike QR method.

From the numerical results obtained for KSPF, it can be concluded that Quasi-Newton method requires some matrix preconditioning techniques to accelerate the convergence.

The numerical results obtained show that the spectral transformation and Arnoldi's method, when applied directly on the system matrix can give selected eigenvalues in

less time than the other selective computation techniques and the QR method. Using the augmented system of equations, for sparsity oriented implementation, takes relatively more time than the direct use of system matrix.

6.2 Future Scope of the Work

1. In each iteration of the Newton Raphson power flow a set of linear equations has to be solved. Instead of using Gaussian elimination method, solution using GMRES method has to be checked for very large systems.
2. The performance of Arnoldi's method in eigenvalue computation has to be evaluated for very large systems represented with detailed models.

REFERENCES

[1] L. Shi, N. Chang, D. Zhao, H. Zhao, H. Zhou, P. T. Tam, Y. Ni, and F. F. Wu, "Implementation of a power system dynamic security assessment and early warning system," in *Power Engineering Society General Meeting, 2007. IEEE*. IEEE, 2007, pp. 1–6.

[2] D.-H. Kuo and A. Bose, "A generation rescheduling method to increase the dynamic security of power systems," *Power Systems, IEEE Transactions on*, vol. 10, no. 1, pp. 68–76, 1995.

[3] K. Sun, S. Likhate, V. Vittal, V. S. Kolluri, and S. Mandal, "An online dynamic security assessment scheme using phasor measurements and decision trees," *Power Systems, IEEE Transactions on*, vol. 22, no. 4, pp. 1935–1943, 2007.

[4] R. Tiako, D. Jayaweera, and S. Islam, "A class of intelligent algorithms for on-line dynamic security assessment of power systems," in *Universities Power Engineering Conference (AUPEC), 2010 20th Australasian*. IEEE, 2010, pp. 1–6.

[5] Y. Saad, "Krylov subspace methods for solving large unsymmetric linear systems," *Mathematics of computation*, vol. 37, no. 155, pp. 105–126, 1981.

[6] Y. Saad and M. H. Schultz, "Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on scientific and statistical computing*, vol. 7, no. 3, pp. 856–869, 1986.

[7] H. Nam, S. Song, K. Shim, D. Kim, Y. Moon, and C. Moon, "Hessenberg method for small signal stability analysis of large scale power systems," in *Power Engineering Society Winter Meeting, 2000. IEEE*, vol. 2. IEEE, 2000, pp. 872–876.

[8] M. Pai, D. S. Gupta, and K. Padiyar, *Small signal analysis of power systems*. Alpha Science Int'l Ltd., 2004.

[9] I. J. Pérez-Arriaga, G. C. Verghese, and F. C. Schweppe, "Selective modal analysis with applications to electric power systems, part i: Heuristic introduction," *Power Apparatus and Systems, IEEE Transactions on*, no. 9, pp. 3117–3125, 1982.

[10] M. Pai and H. Dag, "Iterative solver techniques in large scale power system computation," in *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, vol. 4. IEEE, 1997, pp. 3861–3866.

[11] A. Semlyen, "Fundamental concepts of a krylov subspace power flow methodology," *Power Systems, IEEE Transactions on*, vol. 11, no. 3, pp. 1528–1537, 1996.

[12] L. Wang and A. Semlyen, "Application of sparse eigenvalue techniques to the small signal stability analysis of large power systems," *Power Systems, IEEE Transactions on*, vol. 5, no. 2, pp. 635–642, 1990.

[13] M. H. Gutknecht, "A brief introduction to krylov space methods for solving linear systems," in *Frontiers of Computational Science*. Springer, 2007, pp. 53–62.

[14] W. E. Arnoldi, "The principle of minimized iterations in the solution of the matrix eigenvalue problem," *Quart. Appl. Math*, vol. 9, no. 1, pp. 17–29, 1951.