

FUNDAMENTAL CONCEPTS OF A KRYLOV SUBSPACE POWER FLOW METHODOLOGY

Adam Semlyen

Department of Electrical and Computer Engineering
University of Toronto
Toronto, Ontario, Canada M5S 1A4

Abstract -The well established power flow methods – Gauss-Seidel, Newton-Raphson, and the Fast Decoupled Load Flow – are all based on major, classical methodologies of applied mathematics. The Krylov Subspace Power Flow (KSPF) presented in this paper uses a newer, very successful approach – the Krylov subspace methodology – developed in applied linear algebra for the iterative solution of large, sparse systems of linear equations. The method has been adapted to nonlinear equations and used for the solution of the power flow problem with either an approximation of the Jacobian, as in the Fast Decoupled Load Flow, or in a direct Newton-like manner but without explicitly forming the Jacobian. Convergence rates are from linear to almost quadratic. First the general methodology is described in the paper and then its application to the power flow problem. The main advantage of KSPF is that no matrix factorizations, only sparse matrix-vector multiplications or evaluations of residuals are used. Preliminary tests suggest that KSPF may become a competitive alternative to existing methods, especially in the case of large systems.

1 INTRODUCTION

Power flow programs are probably the most fundamental and widely used tools in the analysis of power systems. They have accordingly been developed to a very high level of perfection. The main methodologies used are: the Gauss-Seidel, the Newton-Raphson, and the Fast Decoupled Power Flow (developed by Stott and Alsac [1]). For special system configurations, special programs can be used to advantage (e.g., for radial and for weakly meshed systems [2,3]). These well established solution methodologies have evolved so as to eliminate some of the shortcomings of previously used methods: the Gauss-Seidel approach is very simple but has slow convergence; the Newton-Raphson method converges quadratically but requires repeated Jacobian evaluations and factorizations; finally, the Fast Decoupled Load Flow does not require any Jacobian evaluation and does not require refactorization of the iteration matrices, but has only linear convergence. Are there any further improvements possible?

A common feature of both the Newton-Raphson and the Fast Decoupled methodologies is that large, sparse matrices have to be factorized for the direct solution of linear equations. However, these equations can also be solved iteratively and there are several very successful methods used to achieve this efficiently, without factorizations, using only matrix-vector multiplications [4,5].

95 SM 600-7 PWRs A paper recommended and approved by the IEEE Power System Engineering Committee of the IEEE Power Engineering Society for presentation at the 1995 IEEE/PES Summer Meeting, July 23-27, 1995, Portland, OR. Manuscript submitted December 22, 1994; made available for printing June 20, 1995.

They seem to have developed in part by the generalization of the conjugate gradient method to nonsymmetric systems and by inspiration from Lanczos-Arnoldi type methods used in the solution of large eigenvalue problems [6]. The latter are closely related to the power method and simultaneous iterations which result in operations on Krylov sequences or subspaces. There exists thus a basis for developing solution methods for the Power Flow problem in power systems, such that factorizations could be completely avoided. Moreover, the calculation of a Jacobian matrix is also not needed just as in the Fast Decoupled Load Flow [1]. The resultant advantage in speed and efficiency is expected to be important in the case of large systems.

In a recent study, Galiana, et al [7] have successfully applied a pre-conditioned conjugate gradient algorithm to the solution of the DC and the Fast Decoupled Load Flow problems, with contingency analysis. These load flow methods, in contrast to the Newton-Raphson Power Flow, satisfy the requirement of the solution procedure for a symmetrical, positive definite, system matrix. As expected, the efficiency of the methodology increases with the size of the system.

The purpose of the present paper is to examine the possibility of using the new, advanced, iterative methods for the practical solution of the load flow problem. The approach developed will be called the *Krylov Subspace Power Flow* (KSPF). Since, however, the concept and methodology of Krylov subspace procedures is not widely known, the derivations in this paper will be fairly complete yet simple enough so that the main ideas are not obscured by details.

In the following, first a general method for solving a *nonlinear* system of equations by Krylov subspace approximations is derived, then the application of this methodology to the power flow problem is discussed. Two approaches will be described and evaluated: the first, with a constant-matrix approximation of the Jacobian, much as in the Fast Decoupled Load Flow, the second, with Newton-like accurate linearization but without explicit calculation of the Jacobian matrix.

2 KRYLOV SUBSPACE METHODOLOGY

The Krylov subspace methods [8,9] 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, nonsymmetric A -matrices. Usually the approach was to *minimize the residual r in the formulation $r = b - Ax$* . This led to approaches with names such as MINRES, ORTHOMIN, ORTHODIR, ORTHORES, and GMRES [4,5]. GMRES [10] is the most widely known and used. The power flow problem is however nonlinear. Since this in any case implies an iterative approach, it is natural to simply extend the iterations of the Krylov subspace minimal residual methodology to the solution of nonlinear equations. The basic ideas are presented in the next section.

2.1 Solving $f(x) = 0$

The power flow equations can be represented in the general form

$$f(x) = 0 \quad (1)$$

where $x (\in R^N)$ represents voltages and phase angles and $f(\cdot) (\in R^N)$ is the difference of the calculated and specified powers, real and reactive. At iteration step k , (1) gives the residual

$$r_k = f(x_k) \quad (2)$$

The linearized form of (2) is

$$r_k = b - Ax_k \quad (3)$$

Here, strictly speaking, the coefficients b and A vary with k but in the proximity of the solution they can be considered constant. In particular, A would then represent the converged Jacobian. We could regard (3) as an equivalent of (2) for casting the nonlinear problem in a form used for the derivation of Krylov subspace methods for the solution of linear systems. Since in reality the Jacobian varies with the iteration point k , its calculation and recalculation would represent a computational burden similar as in Newton's method. Moreover, since it would be used in a Krylov subspace approach, it would not lead to truly quadratic convergence. Therefore, in one alternative, it is reasonable to assume that A is only an approximation of the Jacobian, possibly symmetrical, and affording a decoupling similar to that in the Fast Decoupled Load Flow (FDLF). In a second alternative, the Quasi-Newton approach, A is gradually approaching the Jacobian matrix but is not calculated explicitly. We still use it in the presentation of the general methodology.

In the following derivation we shall use both (2) and (3) for obtaining the minimal residual solution of problem (1): equation (3) to obtain a Krylov subspace update Δx for x_k , as shown below, and (2) to obtain a new, minimal residual, so that x_k is driven to the true solution of (1).

The iterative process starts with a guess x_0 and (2) gives r_0 . At step k we know x_k and (2) yields r_k . Let

$$x_{k+1} = x_k + \Delta x \quad (4)$$

be the new value of the vector of unknowns. We would like to choose Δx so that the new residual, r_{k+1} , results minimal. To solve this minimization problem, we use approximation (3), written for $k+1$, together with (4):

$$r_k - A\Delta x = r_{k+1} \rightarrow \min \quad (5)$$

or

$$A\Delta x \equiv r_k \quad (6)$$

Actually, (6) could be solved exactly if we were prepared to factorize A . Since we wish to avoid the latter, we just write

$$\Delta x = A^{-1}r_k \quad (7)$$

and do the following, somewhat heuristic, reasoning. It is basic to the Krylov subspace approach and perhaps the only conceptually more difficult part of the entire methodology.

The matrix A^{-1} in (7) is an analytical function of A and therefore could be expressed as a series in terms of the powers of A . Since (by the Cayley-Hamilton theorem [11]) A satisfies its own characteristic equation, therefore powers of A equal to or higher than N can be expressed in terms of powers of order less

than N . Thus A^{-1} can be expressed by a polynomial of degree $N-1$. Correspondingly, $A^{-1}r_k$ of (7) is a linear combination of column vectors $A^j r_k$, for $j=0,1,\dots,N-1$. If these vectors are arranged in a matrix, we obtain the Krylov matrix K of order N . Often this matrix is poorly conditioned since r_k may be close to the span of only a part of the eigenvectors of A . (It is clear that, if r_k is proportional to an eigenvector of A , then K has only one independent column, since the first two, r_k and $A r_k$, are linearly related; see [12] page 36.) Therefore new columns added to K become increasingly redundant and only the first columns are essential in the solution. This of course is only an approximation but it leads to computational efficiency. Thus if only $m \ll N$ columns of $A^j r_k$ (for $j=0,1,\dots,m-1$) are used, we have the Krylov matrix $K = [r_k \ A r_k \ A^2 r_k \ \dots \ A^{m-1} r_k]$ of order m which, if of full rank, spans a subspace of order m ($\ll N$). The linear combination of the columns of K is then Ky (with $y \in R^m$).

Since K does not span the full space R^N , (7) can be only approximately satisfied and we set instead:

$$\Delta x = Ky \quad (8)$$

Thus, we simply assume that Δx is confined to the Krylov subspace, the span of K .

Consequently, substitution of Δx from (8) into (5) yields

$$r_k - AKy = r_{k+1} \rightarrow \min \quad (9)$$

where AK is itself a Krylov matrix ($AK = [Ar_k \ A^2 r_k \ \dots \ A^m r_k]$).

To solve the least squares problem (9) for y , we could first be tempted to use a QR decomposition of AK . We note however that the obtained y , when substituted into (8) (in order to calculate Δx), has to be premultiplied by K , so that the total amount of computations would be large. Therefore, for solving the least squares problem (9), we shall use a *special minimization procedure*, based on an Arnoldi-type orthogonalization algorithm (used in GMRES [10]), which is computationally very efficient. Details are given in the next section.

Arnoldi's method [13] gives the orthogonalization of a Krylov matrix K to become the orthonormal matrix V . Then equations (8) and (9) take the form

$$\Delta x = Vz \quad (10)$$

$$r_k - AVz = r_{k+1} \rightarrow \min \quad (11)$$

With the result z of the least squares solution of (11), we obtain Δx from (10). Subsequently (4) yields x_{k+1} and from (2) we obtain r_{k+1} which of course is not exactly the same as the residual from (11). It is this value that will be used in the next iteration.

Because of the constraint we have imposed on Δx , we cannot of course expect for the new residual r_{k+1} to be zero. It will however be orthogonal to the Krylov subspace spanned by AK (or AV) and it will be minimal. Thus, in the case of a linear problem (3), the convergence will be monotonic and, consequently, the approach is robust. The descending pattern will be assured even for the nonlinear problem (2) as long as its approximation (3) is sufficiently close.

2.2 Least Squares Solution of (11)

The transition from formulation (9) to (11) of the minimum residual problem is achieved by a Gram-Schmidt orthogonalization with a special feature, Arnoldi's algorithm. We

present below Arnoldi's method corresponding to the classical Gram-Schmidt orthogonalization. We note however that sometimes an implementation based on the Modified Gram-Schmidt algorithm [14] is preferable.

In the following, the matrix K is represented by the pair: v_1, A , where

$$v_1 = r_k / \|r_k\| \quad (12)$$

becomes the first vector in V . The other vectors v_i are obtained as shown below. Note that the special feature of Arnoldi's method is that the new vector to be orthogonalized (in (13b)) is not taken from the columns of matrix K , but is expressed as Av_j , in terms of the last vector, v_j , brought into the basis. This has important useful consequences.

ALGORITHM ARNOLDI

For $j = 1, 2, \dots, m$, calculate:

$$h_{i,j} = (Av_j, v_i), \quad i = 1, 2, \dots, j \quad (13a)$$

$$\hat{v}_{j+1} = Av_j - \sum_{i=1}^j h_{i,j} v_i \quad (13b)$$

$$h_{j+1,j} = \|\hat{v}_{j+1}\| \quad (13c)$$

$$v_{j+1} = \hat{v}_{j+1} / h_{j+1,j} \quad (13d)$$

Equation (13b), together with (13c) and (13d), can be written as

$$Av_j = \sum_{i=1}^{j+1} h_{i,j} v_i \quad (j = 1, 2, \dots, m) \quad (14)$$

or in the compact matrix form:

$$AV = V_{aug} H \quad (15)$$

where V_{aug} is the matrix V augmented by the new column v_{m+1} , and H is a rectangular upper Hessenberg matrix (of dimension $(m+1) \times m$) with elements $h_{i,j}$. Note that the last row of H consists of the single element $h_{m+1,m}$.

The importance of (15) is that it permits to replace AV in (11) by $V_{aug} H$. The result is

$$r_k - V_{aug} H z = r_{k+1} \rightarrow \min \quad (16)$$

Now, to minimize the new residual, r_{k+1} , we note that it must be orthogonal to the span of $V_{aug} H$. Therefore, we premultiply (16) by $(V_{aug} H)^T$ and set the result equal to zero. This yields, taking into account the orthogonality of V_{aug} ,

$$H^T H z = H^T V_{aug}^T r_k \quad (17)$$

which, clearly, is the normal equations form for the least squares solution of the problem

$$H z = V_{aug}^T r_k \quad (18)$$

Taking into account (12), where for notational convenience we write $\beta = \|r_k\|$, (18) becomes

$$H z = b \quad (19)$$

where $b = [\beta, 0, 0, \dots, 0]^T$ has length $m+1$. A slightly different derivation of (19) is given in [10].

The next step is to solve the small size least squares problem

(19) for z , required for obtaining Δx from (10). Again, the QR decomposition is reasonable for this purpose. Details for its efficient implementation are given in [10]. In conclusion, the simple and low order equation (19) replaces (11) in the general solution process.

An important remark is in place at this stage. When A represents the Jacobian, it is nonsymmetrical. However, when the effect of phase shifters is not included in A , then, in approximating the network susceptance matrix, A is symmetrical and the Modified Gram-Schmidt version of Arnoldi becomes the Lanczos algorithm [14] (see the Appendix). Then, of course, the upper Hessenberg matrix H reduces to symmetrical tridiagonal form and, instead of the summation in (13b), we have only a short, three-term, recursion. Thus, increasing the order m of the orthogonalized Krylov subspace does not entail more than proportional computing and memory requirements. Loss of orthogonality [15] is not of serious concern in our application since after m steps a new residual is calculated and the Arnoldi/Lanczos process is restarted. We also note that the Lanczos method is closely related to the method of Conjugate Gradients [5,15].

In the following – in order to simplify the presentation – we shall not point directly to the symmetrical Lanczos implementation of the Krylov subspace orthogonalization but will refer in all cases, whether symmetrical or nonsymmetrical, to the more general Arnoldi approach. In the symmetrical case, Lanczos is implied.

2.3 Solution Strategies

The basis of the whole Krylov subspace methodology described above is clearly the Arnoldi algorithm which gives both V and H needed in (10) and (19). At each iteration k , we choose the order m of the Krylov subspace, so that m is a function of k . Possibly, m is a constant or, perhaps, $m = k$ or proportional to k up to $m_{\max} \ll N$. In general, m could be chosen adaptively as the computation progresses, with some fixed value m_i for each set of n_i ($i = 1, 2, \dots$) iterations. The issues of convergence are very complex. They are analyzed to great depth in [5] and in the monograph [16] (see also [17]).

The solution strategy will depend on the nature of the problem to be solved which can be best characterized by the spectrum (the totality of the eigenvalues) of the system matrix A . Best and fastest convergence is obtained, in descending order, for A being:

- symmetric (all eigenvalues are real) and definite
- symmetric indefinite
- nonsymmetric (complex eigenvalues may exist in conjugate pairs) and definite real (definiteness relates to real part), i.e. the symmetric part $(A + A^T)/2$ is positive or negative definite
- nonsymmetric general

In the power flow problem (see Section 3) the Jacobian matrix is in general of type (c) above but the FDLF-approximation to the Jacobian is in general of type (a).

2.4 General Methodology

We review below the general solution procedure based on the equations developed in the previous sections.

- Formulate the nonlinear equation (2) to be used for the calculation of the residual r_k and choose the approximation A

to be used in the sequel in the minimization of the residual. Adopt an iteration strategy $m(k)$, as discussed above.

- (b) Initialize: choose x_0 , obtain r_0 from (2).
- (c) Iterate. For $k=1,2,\dots$ until satisfied ($r_{k+1} < \text{tolerance}$):
- ◆ Do ARNOLDI: obtain $\beta = \|r_k\|$, V , and H ;
 - ◆ Solve (19) for z , obtain Δx from (10) and x_{k+1} from (4);
 - ◆ Calculate r_{k+1} from (2).

2.5 Implementation Alternatives

The essential difference between our problem of solving nonlinear equations (1) and the underlying existing methodology for solving the linear system $Ax=b$ is that in our case, as emphasized from the beginning, matrix A is only an approximation of the Jacobian. Therefore, the least squares solution of Section 2.2 is only approximate and the solution process may require in a constant-matrix alternative a good number of iterations. On the other hand, since A is only an approximation of what it ideally could be, the least squares problem does not necessarily have to be solved accurately if the alternative is computationally advantageous. We also add the remark that the calculation of the residual (2) for the nonlinear problem is computationally more expensive than the corresponding computation (3) for a linear problem. Consequently, the following options have to be examined when deciding on details and strategies of implementation.

- (a) What is preferable:
- The *constant-matrix alternative*, with faster but more numerous, linearly converging, iterations, or
 - The *quasi-Newton alternative*, with longer but fewer, almost quadratically converging iterations? Clearly, the latter is advantageous when higher accuracy is essential.
- (b) Should (19) be solved as a least squares problem? The alternative is to set in (15) the last element of H equal to zero (it becomes a regular $m \times m$ Hessenberg matrix) and then V_{aug} becomes V , and b in (19) will have length m . Thus (19) becomes a simple (small) set of linear equations. Some preliminary tests have shown that this simplification has actually improved the rate of convergence.

Answers to these questions and recommendations will be given in Section 4.

3 THE POWER FLOW PROBLEM

The power flow problem consists in the solution of nonlinear equations as (1) where x stands for the unknown voltage magnitudes V_i and their phase angles θ_i , and the residuals in (2) represent mismatches of active and reactive powers, P_i and Q_i . We have thus the two sets of equations, for $i=1,2,\dots, j=1,2,\dots$,

$$P_i(\theta, V) = 0 \quad \text{and} \quad Q_j(\theta, V) = 0 \quad (20)$$

In the two following sections we shall examine the options in KSPF for the solution of the power flow problem: first the *constant-matrix alternative* and then the *quasi-Newton alternative*.

3.1 The Constant-Matrix Alternative

When linearized, the power-voltage equations (20) take form (3), where A is the Jacobian matrix

$$J = \begin{bmatrix} \partial P / \partial \theta & \partial P / \partial V \\ \partial Q / \partial \theta & \partial Q / \partial V \end{bmatrix} \quad (21)$$

Note that matrix A can be modified by changing the *sequence of the variables* in (20) and (21). Thus we can affect the spectrum of A (discussed in Section 2.3) and, with it, the convergence properties of the iterative process in the Krylov Subspace Power Flow (KSPF). The sequence (21), which permits the decoupling in the Fast Decoupled Load Flow, is also optimal in the sense that (21) is almost symmetrical.

When approximated by a constant matrix as in the Fast Decoupled Load Flow, the Jacobian matrix (21) becomes the pair of the familiar B -matrices. These are then used for A in (3) and in the resulting KSPF.

3.2 The Quasi-Newton Alternative

In the constant-matrix alternative discussed in the previous section, A is an approximation of the Jacobian, as in the Fast Decoupled Load Flow. It is used in the computation of the product Av_j , with the j th vector v_j of the Krylov matrix K , in (13a) and (13b) of the Arnoldi algorithm. An improvement could be expected if instead of a constant A we would use the Jacobian itself, but this would be computationally impractical. We note, however, that A is not needed explicitly, since it is only used as an operator to obtain the vector Av_j . The latter can be calculated directly as the directional derivative of $f(x)$ in (1), or of the power functions $p(x)$ in

$$f(x) = p_{sp} - p(x) \quad (22)$$

(where p_{sp} represents the specified active and reactive powers), since the increments of the two functions are the same, except for the sign. Thus, if we linearize $p(x)$ at $x_k + \varepsilon v_j$, around x_k (with the small increment εv_j), we get

$$p(x_k + \varepsilon v_j) = p(x_k) + J_k \varepsilon v_j \quad (23)$$

This yields, with the Jacobian J_k replaced by A ,

$$Av_j = \frac{p(x_k + \varepsilon v_j) - p(x_k)}{\varepsilon} \quad (24)$$

This is the directional derivative evaluation of the product Av_j . It requires a single (vector) function evaluation (of $p(x_k + \varepsilon v_j)$, in addition to the already available $p(x_k)$).

Thus a Newton-like solution can be obtained, the closer to the true Newton's method the larger the dimension m chosen for the dimension of the Krylov subspace. If m is large enough, an almost quadratic convergence pattern can be expected. This is of course obtained at the price of an increased number m of function evaluations (equivalent to the calculation of the residual vector). We also note here again that, while basically a sparse computation, a function evaluation is more expensive than a sparse matrix-vector multiplication because of the trigonometric functions involved and also because the real part, G , of the admittance matrix is not neglected as in the constant-matrix approach (or FDLF).

4 TEST RESULTS

4.1 Illustration of Different Solution Alternatives

In Section 2.5 several alternatives for solving the general system (1) of nonlinear equations using a Krylov subspace methodology have been formulated. In the following, the effect of the adopted strategy will be illustrated on hand of convergence plots for the (infinite norm of the) residual of a small system (20-buses: 5 generator and 15 load buses). All tests were performed on a 486-DX2 66 MHz PC using MATLAB v.4.0 which has sparse matrix capabilities [18].

Figures 1 and 2 show the *constant-matrix alternative* in terms of the size m of the Krylov subspace. Figure 1 gives the convergence for the least squares implementation while Figure 2 corresponds to the simplified solution, as detailed at (b) in Section 2.5. Clearly, after a faster initial stage, the convergence becomes linear and steady, down to almost machine precision. In the case of the simplified solution, the monotonic decrease of the residual is not assured (see the plot for $m=3$ in Figure 2) but it is still present in most cases; also, the rate of convergence is less steady. In both implementations it can be seen that, at first, convergence becomes faster if m is increased but *there exists an optimal value for m* beyond which the rate of convergence slows down. The optimal size m of the Krylov subspace is small (6, respectively 5, in the two implementations) but in terms of computing time a smaller m , say 4 is a better choice. With the simplified solution (as opposed to least squares) the convergence rate improves. Thus the *simplified approach appears to be preferable* to the full, least squares implementation.

It is interesting to note that the computing time in the cases of both Figures 1 and 2 (in the range of 20-22 seconds) increased only slightly with m and was almost identical to that for FDLF.

Figure 3 illustrates the convergence of the *quasi-Newton approach* in terms of the parameter m . We see that the convergence is *much faster* than in the constant-matrix case but, of course, at the higher cost of additional (vector) function evaluations needed in (24). Only if m is significantly increased does the super-linear convergence become apparent and significant. The computer time, for increasing m , from 8 to 12 and 18, was of 19, 43, and 62 seconds, respectively.

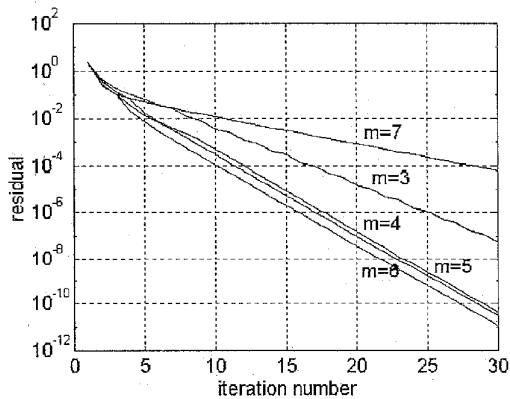


Figure 1 Constant matrix alternative, least squares solution

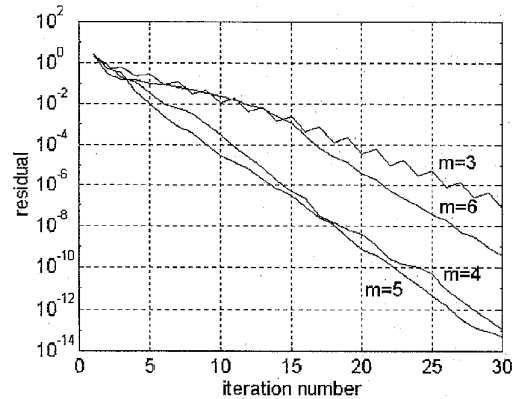


Figure 2 Constant matrix alternative, simplified approach

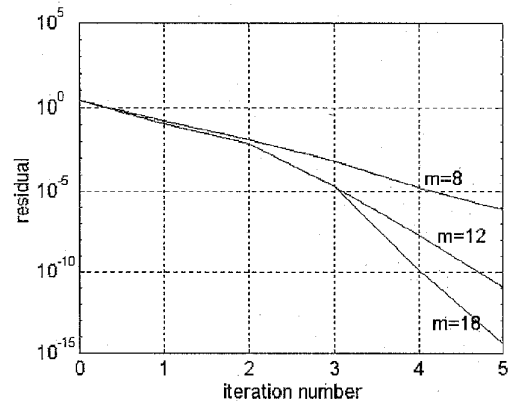


Figure 3 Quasi-Newton alternative

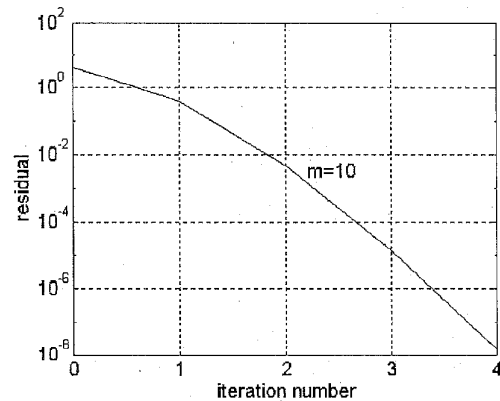


Figure 4 Convergence of KSPF for a 57 bus system

Figures 1 to 4 show the residuals down to very small values in order to portray the rates and patterns of convergence for both the Constant Matrix and the Quasi-Newton alternatives. However, for

practical calculations, residuals smaller than 10^{-4} will usually not be needed. In particular, when operating conditions, such as reactive power limits at generators or tap settings of transformers, lead to modifications in the load flow equations, these changes can be implemented at higher residual levels. The same applies to repeated load flow solutions, as in contingency analysis. In all these situations, a modified load flow can be initiated without of course the burden of refactorization or of additional computations related to the use of the Matrix Modification Lemma.

4.2 Characterization of Solution Alternatives

Several modest size test systems have been used to validate the feasibility of the two basic alternatives of KSPF including the 14, 30, 57, and 118-bus test systems described in [19]. Some have been obtained from randomly generated network topologies and admittances (within ranges of values as in real systems). The acceptance of the latter as adequate for the assessment of a power flow methodology is based on the spectrum of the B -matrix which is similar to that of the real systems.

We note first that KSPF, by using a lower order (Krylov subspace) approximation of the Jacobian matrix or even of its B -matrix approximation used in FDLF is not expected to necessarily have the same robustness as the classical power flow methods. Its behavior is also more difficult to analyze and predict, as the underlying methodology is that of iterative solution of linear equations as opposed to direct methods. The expected advantage is in the possible savings in computing time by completely obviating the need for matrix factorizations. The experiments performed confirmed these expectations with the following detailed observations:

- (a) The Quasi-Newton KSPF was the most reliable since it is based on a variable A -matrix that approximates the Jacobian. As in Newton's method, its convergence requires satisfactory initial values. Figure 4 shows the convergence with a low order Krylov subspace ($m = 10$) of a 57 bus system (18 generator, 39 load buses). The number of iterations is similar as in a true Newton-Raphson procedure, achieved however with lower order matrices and without factorizations.
- (b) The Constant Matrix KSPF proved to be less robust due to the triple approximation: Jacobian to B -matrix, the B -matrix to a low order Krylov matrix (see Figures 1 and 2) and, especially at the beginning, the linearization itself. The systems where the Constant Matrix KSPF failed to converge were, in a simple characterization, those with weak, sparse, paths from the generators to the loads. This dependence became apparent by adding or removing existing links in the network. It is felt that improvements can be made to the Constant Matrix KSPF when its working will be better understood.
- (c) The convergence of the Constant Matrix KSPF is much slower than in the case of the Quasi-Newton approach and the smaller size m of the Krylov matrices in the former does not compensate for this fact (see Figures 1 and 2). However, in the latter, the calculation of the directional derivatives of (24) requires vector function evaluations with analytical calculations akin to those in the computation of residuals.

These take significantly more time than the calculation of sparse matrix-vector products. Therefore, the total computing time, for both the Constant Matrix and the Quasi-Newton alternatives, has resulted of approximately equal order, on the average, with variations depending on the systems and on the strategies adopted, in particular for choosing m . Moreover, the computing time for FDLF, with all programs in MATLAB implementation, also resulted of the same order.

5 CONCLUSIONS

The paper has presented the fundamental ideas of a new power flow methodology, based on the use of approximations in a Krylov subspace [10].

The essential feature of the Krylov Subspace Power Flow (KSPF) is that the linear equations that appear in the iterative process are not solved directly and accurately by Gaussian elimination as in the Newton-Raphson and the Fast Decoupled Load Flow. The indirect solution methodology of KSPF is of the same class as that of the Gauss-Seidel power flow but has better convergence characteristics. In an alternative using a constant and "decoupled" matrix as an approximation to the Jacobian, KSPF has somewhat similar features as the Fast Decoupled Load Flow. In a second alternative which approximates the Jacobian matrix without ever calculating it, the quasi-Newton option of KSPF behaves similarly to the Newton-Raphson power flow, including its convergence pattern which is close to quadratic.

The procedures for the indirect solution of large, sparse systems of linear equations are of an extraordinary variety and multiplicity [4,5]. Therefore, having examined the application of one particular iterative approach, basically a variant of the widely recognized and popular GMRES [10], to the solution of the nonlinear equations of the power flow problem, does not justify drawing at this stage definitive conclusions about the competitive value of KSPF.

It is however clear that, since at each iteration in KSPF the solution obtained is only approximate, more iterations can generally be expected in KSPF than with the corresponding direct solution method. On the other hand, since in KSPF there are no matrix factorizations as in traditional power flows (with the unavoidable fill-ins even at the best ordering and correspondingly denser L, U factors), it is clear that the sparse matrix-vector multiplications in KSPF are an essential contributor to its efficiency. This is of particular significance in larger systems. Moreover, since in KSPF large matrix factorizations are not needed, it can potentially be applied to advantage for repeated load flow solutions in contingency analysis.

KSPF is a first application of the Krylov subspace methodology to the solution of power system problems. The methodology may well have other applications worthwhile to be explored, for instance, to optimal power flow solutions or state estimation.

6 ACKNOWLEDGMENTS

Financial assistance by the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

7 APPENDIX The Symmetric Lanczos Method

As mentioned, H obtained in the Arnoldi process becomes a tridiagonal matrix. Its diagonal elements are now denoted by α_j (for $h_{j,j}$) and those on the side-diagonals by β_j (for $h_{j,j+1} = h_{j+1,j}$). Set $\beta_0 = 0$, $v_0 = 0$.

ALGORITHM LANCZOS

For $j = 1, 2, \dots, m$, calculate:

$$w_j = Av_j - \beta_j v_{j-1} \quad (25a)$$

$$\alpha_j = (w_j, v_j) \quad (25b)$$

$$\hat{v}_{j+1} = w_j - \alpha_j v_j \quad (25c)$$

$$\beta_j = \|\hat{v}_{j+1}\| \quad (25d)$$

$$v_{j+1} = \hat{v}_{j+1} / \beta_j \quad (25e)$$

8 REFERENCES

- [1] B. Stott and O. Alsac, "Fast Decoupled Load Flow", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-93, No. 3, May/June 1974, pp. 859-869
- [2] G.X. Luo and A. Semlyen, "Efficient Load Flow for Large Weakly Meshed Networks", IEEE Transactions on Power Systems, Vol. 5, No. 4, November 1990, pp. 1309-1316
- [3] C.S. Cheng and D. Shirmohammadi, "A Three Phase Power Flow Method for Real Time Distribution System Analysis", IEEE paper no. 94 SM 603-1 PWRs, presented at the 1994 PES Summer Meeting, San Francisco, California
- [4] W. Hackbusch, "Iterative Solution of Large Sparse Systems of Equations", Springer-Verlag, 1994
- [5] O. Axelsson, "Iterative Solution Methods", Cambridge University Press, 1994
- [6] Y. Saad, "Numerical Methods for Large Eigenvalue Problems", Halsted Press, 1992
- [7] F.D. Galiana, H. Javidi, and S. McFee, "On the Application of a Pre-conditioned Conjugate Gradient Algorithm to Power Network Analysis", IEEE Transactions on Power Systems, Vol. 9, No. 2, May 1994, pp. 629-636
- [8] R. F. Freund, G.H. Golub, and N.M. Nachtigal, "Iterative Solution of Linear Systems", Acta Numerica, 1991, pp. 57-100
- [9] Y. Saad, "Krylov Subspace Methods for Solving Large Unsymmetric Linear Systems", Mathematics of Computation, Vol. 37, No. 155, 1981, pp. 105-126
- [10] Y. Saad and M.H. Schultz, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems", SIAM J. Sci. Stat. Comput., Vol. 7, No. 3, 1986, pp. 856-869
- [11] D.S. Watkins, "Fundamentals of Matrix Computations", John Wiley & Sons, 1991
- [12] J.H. Wilkinson, "The Algebraic Eigenvalue Problem", Clarendon Press, Oxford, 1965
- [13] W.E. Arnoldi, "The Principle of Minimized Iteration in the Solution of the Matrix Eigenvalue Problem", Quart. Appl. Math., Vol. 9, 1951, pp. 17-29
- [14] A. Jennings and J.J. McKeown, "Matrix Computation", 2nd edition, John Wiley & Sons, 1992
- [15] G.H. Golub and C.F. Van Loan, "Matrix Computations", 2nd edition, The Johns Hopkins University Press, 1989
- [16] O. Nevanlinna, "Convergence of Iterations for Linear Equations", Birkhäuser Verlag, 1993
- [17] M. Hanke : Review of [16], in SIAM Review, Vol. 36, No. 3, September 1994, pp. 508-509
- [18] J.R. Gilbert, C. Moler, and R. Schreiber, "Sparse Matrices in MATLAB: Design and Implementation", SIAM J. Matrix Analysis and Applications, Vol. 13, 1992, pp. 333-356
- [19] Y. Wallach, "Calculations and Programs for Power System Networks", Prentice Hall, 1986 (Section 1.6: Example Networks and Notes)

BIOGRAPHY

Adam Semlyen (Fellow, IEEE) was born and educated in Rumania where he obtained a Dipl. Ing. degree and his Ph.D. He started his career with an electric power utility and held academic positions at the Polytechnic Institute of Timisoara, Rumania. In 1969 he joined the University of Toronto where he is a professor in the Department of Electrical and Computer Engineering, emeritus since 1988. His research interests include the steady state and dynamic analysis of power systems, electromagnetic transients, and power system optimization.

Discussion

H. Dağ and F. L. Alvarado (University of Wisconsin-Madison) The author is to be commended for the nicely written timely paper on Krylov subspace method. It is good to see that such a mature method being used in engineering applications. With the increasing dimension of the system and the introduction of highly nonlinear devices such as FACTS, which need to be modelled in greater detail (hence increasing the problem dimension), direct methods based on Gaussian elimination become impractical. Furthermore, direct methods unlike iterative methods for the solution of linear equations are not amenable to parallel processing [D1, D2].

The fundamental concept of the paper is to use an Arnoldi process to solve smaller least squares problems, which in turn solves the set of linear equations. The set of linear equations are obtained from an approximate linearization of the nonlinear set of power flow equations. The proposed method does not truly minimize the residual of the linearized system, since the residual is computed using the set of nonlinear equations. As the author points out, the biggest advantage of the suggested method is that no factorization and/or matrix update is needed. The calculation of the Jacobian is replaced by a nonlinear vector function evaluation for the quasi-newton approach and is not needed for the constant matrix approach. The following are some problems the discussers encountered when trying out the suggested method.

- The success of the proposed method depends on the eigenvalue spectrum of the approximate jacobian. Thus, it is not easy to predict ahead of time how many iterations are required to solve the problem. With direct methods, however, it is easier to predict the storage requirement and approximate cpu time.
- The choice of parameter m , which determines the rank of the Krylov matrix, can be problem dependent. The bigger m is, the larger the storage requirement for each iterations will be.
- The constant matrix alternative works better with a small m , but takes lots of iterations. The quasi-newton approximation requires nonlinear function evaluations. It is not clear whether the proposed method will be competitive alternative with the direct method for large systems.

The discussers would like to have author's comments on the following questions:

- Would there be any advantage if one solves equation (9) by the conjugate gradient method normal equation formulation instead of a least square minimization problem?

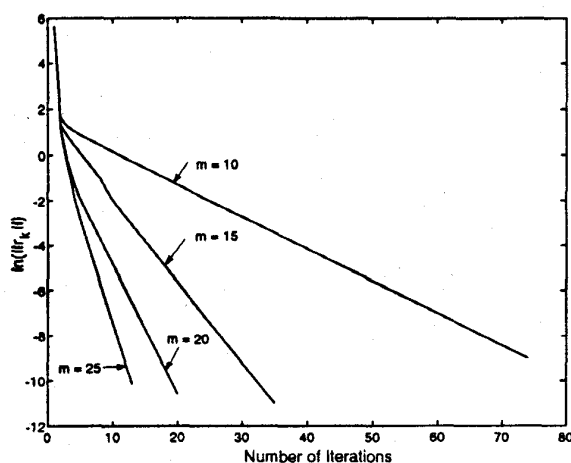


Fig. 1: Minimization of $r_k = b - Ax_k$ for DC load flow matrix B of IEEE 118 bus system using Arnoldi process.

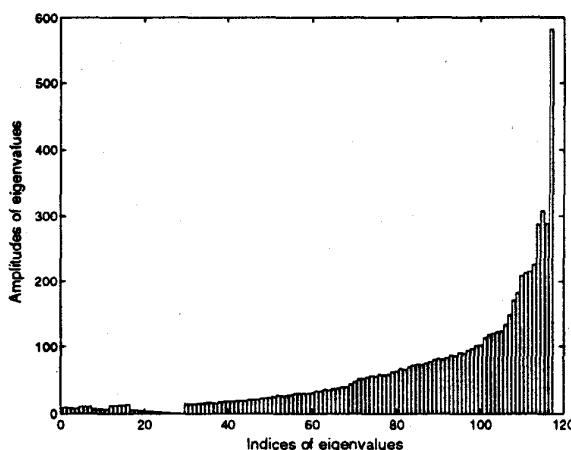


Fig. 2: Eigen spectrum of DC load flow matrix B of 118 bus system.

- Could the author comment on the choice of initial guess x_0 . Specifically, is a flat start as important as it is for the Newton formulation of load flow?
- Is it advantageous to solve equation (19) by using a direct method with last row of H and V removed instead of, for example, using Givens rotations?
- Could the author clarify the following: In Figure 3 of the paper, only when m is as high as 18 (more than half of the system dimension) we see a super linear convergence. This suggests that for the Quasi-Newton alternative m should be fairly big. However, in Figure 4 of the paper we see that the method results in almost super linear convergence with $m = 10$ for the 57-bus system. Is there any correlation between system dimension and m ?

- Results for the Constant matrix alternative (Figures 1-2 in the paper) indicate that in order to obtain a fairly small residual say 10^{-5} one needs about as many iterations as two thirds of number of buses in the system. For very large systems, that many Arnoldi process and least square solves will undoubtedly slow down the solution process. Will the suggested approach be competitive to direct methods or to the recently suggested CG solver in reference [7]?

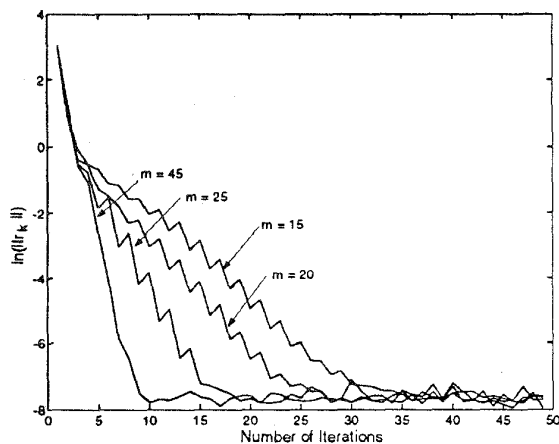


Fig. 3: Constant Matrix approach for 118 bus system.

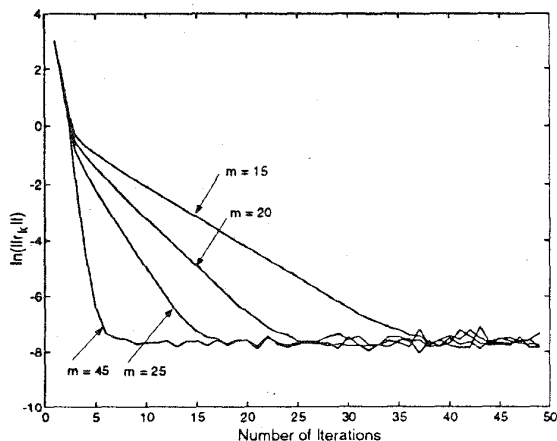


Fig. 4: Quasi-Newton approach for 118 bus system.

One disturbing observation is that our (admittedly rushed) implementation of the suggested algorithm did not converge to tolerances less than 10^{-6} for both of the approaches. The implementation of Arnoldi process to solve a set of linear equations (corresponding to solving for Δx in the paper) is tested using a DC load

flow matrix B for the IEEE 118 bus test system with a random right hand side vector. The matrix B has the eigenspectrum depicted in Figure 2. It is symmetric and positive definite. The method converges monotonically. Furthermore, it takes fewer with larger m , as expected. When the residual is computed from equation (2) of the paper, i.e. $r_k = f(x_k)$, the method does not converge monotonically when using the constant matrix approach with small m (as shown in Figure 3). It does converge monotonically, however, for any choice of m for the quasi-newton approach (as shown in Figure 4). Both Figures 3 & 4 are obtained by the application of the KSPF method to the IEEE 118 bus test system with 54 generators.

REFERENCES

- [D1] H. Dağ and F. L. Alvarado, *Direct Methods Versus GMRES and PCG For Power Flow Problems*, in the proceedings of NAPS, pp.274-278, october 5-9, 1993, Washington, DC.
- [D2] F. L. Alvarado, H. Dağ and M. ten Bruggencate, *Block-Bordered Diagonalization and parallel iterative solvers*, in proceedings of Colorado Mountain Conference on Iterative Methods, April 5-9, 1994, Breckenridge, CO.

Adam Semlyen: I wish to thank the discussers for their valuable comments to the paper and for their very useful contributions to the Krylov Subspace Power Flow (KSPF) solution. The additional references they have provided indicate that their interest in the use of indirect methods dates back several years and reference [D1] seems to be the first in relation to power flow methods.

In response to the problems raised, I would like to give next some insight into the geometry of the Krylov subspace methodology used in the paper.

When we solve the linear system $Ax = b$, we note that, in general, the columns of the matrix A span the whole vector-space of b (of dimension n). Thus these columns can be viewed as a basis of coordinates, so that x gives the coordinates of b with respect to this basis. This decomposition is complete. On the other hand, in the Krylov subspace methodology, in (9) we have a matrix $K' = AK$ of lower rank $m (< n)$, so that its m columns can provide only an incomplete basis for the n -dimensional space of the specified vector. Therefore in (9), i.e., in $K'y \approx r_k$, the right hand side vector r_k cannot be fully represented by the m components available at the left hand side of the equation. What is special about the Krylov subspace is (the heuristically justified expectation) that even this relatively

low order decomposition may, if it is optimal (i.e., a least squares solution for y), closely approximate the given vector r_k . (Surely, the validity of this expectation will depend on the particular problem being considered.) However, K' will in general not be as good as the original matrix A , certainly not if A is the Jacobian matrix. Still, if A is only an approximation of the Jacobian (as in the constant-matrix KSPF), K' may be almost equivalent to A . Figure 1 of the discussion illustrates the fact that the larger m the better K' approximates the original system matrix.

I strongly agree with all three findings listed by the discussers and wish to give the following clarifications to their questions.

- Ideally $m \ll n$, and further research, perhaps exploiting the possibility of efficient preconditioning, should ensure that this is the case. While m is expected to increase with n , this increase should be much below linear. This is demonstrated by the tremendous success of GMRES in other fields of application. Then, a direct method (as for instance the one described in Section 2.2 of the paper) is likely to be more efficient for the lower order problem of (9) or (11) than an indirect method.
- The initial guess x_0 is as crucial in KSPF as in the direct Newton's method because, as discussed above, the Jacobian is degraded by using a Krylov subspace (even though its effect in matrix-vector multiplications is obtained accurately).
- The approximate solution of (19) (as an algebraic equation, the so-called simplified approach of the paper) is equivalent to a weighted least squares solution with vanishing weighting of the last equation. It is computationally simpler and, in the experience reflected in Figures 1 and 2 of the paper, it is also slightly more

efficient (for no clear reason) than the regular least squares solution.

- The fact that superlinear convergence requires a larger value for m seems to be related to the approximation $K' \rightarrow J$ improving with m . The correlation between the required m and system size n is apparently system dependent and its details are not yet clear.
- The competitiveness of KSPF with existing direct methods is not yet assured and will depend on further attention paid to this topic of research. It is however a very encouraging fact that in other fields of science and engineering, indirect solution methods are the only practical approach for truly large systems.
- Figures 3 and 4 of the discussion suggest that the tolerances could not be reduced below 10^{-8} because of the finite precision arithmetic (round-off errors). The machine- ϵ in MATLAB is 2.22×10^{-16} and thus a similar lower limit of 10^{-16} exists also for the curves in all figures in the paper.
- I am very pleased to see that the discussers have obtained essentially similar results to mine on the 118 bus IEEE test system. It is my belief that we shall see significant progress in this field in the near future.

Some references to the underlying topic of iterative solution of nonlinear systems of equations are listed below.

- [A] P.N. Brown, "Hybrid Krylov Methods for Nonlinear Systems of Equations", *SIAM J. Sci. Stat. Comput.*, Vol. 11, No. 3, 1990, pp. 450-481.
- [B] C.T. Kelley, "Iterative Methods for Linear and Nonlinear Equations", Society for Industrial and Applied Mathematics, 1995.

Manuscript received October 9, 1995.