CPFLOW: A Practical Tool for Tracing Power System Steady-State Stationary Behavior Due to Load and Generation Variations

Hsiao-Dong Chiang, Alexander J. Flueck
School of Electrical Engineering
Cornell University
Ithaca, NY 14853

Kirit S. Shah
System Planning Department
Union Electric Company
St. Louis, MO 63166

Neal Balu
Power Systems Planning & Operations
Electrical Systems Division
EPRI

Abstract

In this paper, a computer package called CPFLOW which is a comprehensive tool for tracing power system steady-state stationary behavior due to parameter variations is presented. The variations include general bus real and/or reactive loads, area real and/or reactive loads, or system-wide real and/or reactive loads, and real generation at P-V buses (e.g. determined by economic dispatch or participation factor). The main advantages of CPFLOW over repetitive power flow calculations are its computational speed and reliability as well as its wide applicability. A detailed description of the implementation regarding the predictor, corrector, step-size control and parameterizations employed in CPFLOW is presented. CPFLOW has comprehensive modeling capability and can handle power systems up to 12000 buses. For an illustrative purpose, CPFLOW is applied to a 3500-bus power system with a comprehensive set of operational limits and controls.

1. Introduction

CPFLOW (Continuation Power Flow) is a comprehensive tool for tracing power system steady-state behavior due to load and generation variations. In particular, CPFLOW can generate P-V, Q-V, and P-Q-V curves with the capability that the controlling parameter λ can be one of the following

- general bus (P and/or Q) loads + real power generation at P-V buses
- area (P and/or Q) loads + real power generation at P-V buses
- system (P and/or Q) loads + real power generation at P-V buses

CPFLOW, computationally based on the continuation method [1], can trace the power flow solution curve, with respect to any of the above three varying parameters, through the 'nose' point (i.e. the saddle-node bifurcation point) without any numerical difficulty. The continuation methods have been used by several researchers to trace the power flow solution curve [3,4,5,6]. The main differences between the previous work and CPFLOW lie in (i) the level of modeling capability, (ii) the level of applicability (through different schemes of parameterizations), (iii) the computational speed and reliability (through different predictors, correctors and step-size control).

The main advantages of CPFLOW over repetitive power flow calculations are the following:

- **Computation**
  1. it is more reliable than the repeated power flow approach in obtaining the solution curve; especially for ill-conditioned power flow equations
  2. it is faster than the repeated power flow approach via an effective predictor-corrector, adaptive step-length selection algorithm and efficient I/O operations.
- **Function** – it is more versatile than the repeated power flow approach via parameterizations such that general bus real and/or reactive loads, area real and/or reactive loads, or system-wide real and/or reactive loads, and real generation at P-V buses (e.g. determined by economic dispatch or participation factor) can vary.

CPFLOW can be used in a variety of applications such as (1) to analyze voltage problems due to load and/or generation variations (e.g. voltage dip, voltage collapse), (2) to evaluate maximum interchange capability and maximum transmission capability, (3) to simulate power system static behavior due to load and/or generation variations with/without control devices, and (4) to conduct coordination studies of control devices for steady-state security assessment.

CPFLOW uses EPRI IPFLOW [2] as a platform; making its modelling capability quite comprehensive. It can accept power flow data in the format of EPRI, BPA, IEEE, PECO, PTI, etc. The current version of CPFLOW can handle power systems up to 12000 buses.

2. System Model

CPFLOW has the following modelling capability:

- Generators
  - controlling the voltage between bounds
  - reactive power limits
  - real power generations due to participation factors

94 WM 244-4 PWRS A paper recommended and approved by the IEEE Power System Engineering Committee of the IEEE Power Engineering Society for presentation at the IEEE/PES 1994 Winter Meeting, New York, New York, January 30 - February 3, 1994. Manuscript submitted April 21, 1993; made available for printing January 21, 1994.

- Loads
  - constant power, constant current, constant impedance, or combination of them.
  - nonlinear (static load model)
- Switchable shunts and static VAR compensators
- Area interchange schedules
- ULTC transformers
- ULTC phase shifters
- Static tap changers and phase shifters
- DC transmission lines

3. Tracing the Solution Curve

Consider a comprehensive (static) power system model, as described in the previous section, expressed in the following form:

$$0 = f(x,\lambda) = f(x) - \lambda b \tag{1}$$

where $\lambda \in R^1$ is a (controlling) parameter subject to variation and $b \in R^n$ represents the change in real and reactive power load demand and the change in real power generation. Using terminology from the field of nonlinear dynamical systems, system (1) is a one-parameter nonlinear system. In power system applications, a one-parameter dynamical system is a system together with one of the following conditions:

1. the reactive (or real) power demand at one load bus varies; and the real power generations at some collection of generator buses vary, and their variations can be parameterized while the others remain fixed,
2. both the real and reactive power demand at a load bus vary; and the real power generations at some collection of generator buses vary, and their variations can be parameterized. Again the others remain fixed,
3. the real and/or reactive power demand at some collection of load buses varies; and the real power generations at some collection of generator buses vary, and their variations can be parameterized while the others are fixed.

Generally speaking, power systems are dynamical systems and are normally operated near a stable equilibrium point. As system loads and generations change slowly, the stable equilibrium point changes position but remains a stable equilibrium point. This situation may be modelled with the static model (1) by regarding $f(x,\lambda) = 0$ as specifying the position of the stable equilibrium point x as a function of λ. (Here it would be more precise to call $f(x,\lambda) = 0$ a quasi-static model since λ varies and causes corresponding variations in x). This model may also be called a parametric power flow model. Stating it differently, the parameter λ varies slowly or quasi-statically with respect to the dynamics of its counterpart which is the following:

$$\dot{x} = f(x,\lambda) \tag{2}$$

For example, if the system represented is initially near a stable equilibrium point $x_s(\lambda)$, then the dynamics will make the system state track $x_s(\lambda)$ as λ slowly varies.

Exceptionally, variations in λ will cause the stable equilibrium point to bifurcate. The stable equilibrium point may then disappear or become unstable depending on the way in which the parameter is varied and on the specific structure of the system. One typical way in which system (1) may lose stability is that the stable equilibrium point $x_s(\lambda)$ and an unstable equilibrium point $x_1(\lambda)$ coalesce and disappear in a saddle-node bifurcation as parameter λ varies. It is clear that the P-V and Q-V curves commonly used in the power industry to analyze voltage stability and voltage collapse are examples of saddle-node bifurcations.

We next discuss an indirect method to simulate the approximate behavior of the power system (1) due to load and/or generation variation. Before reaching the 'nose' point, the power system with a slowly varying parameter traces its operating point which is a solution of the following equation whose corresponding Jacobian has all eigenvalues with negative real parts:

$$\mathbf{f}(\mathbf{x},\lambda) = 0, \qquad \mathbf{x} \in \mathbf{R}^n, \mathbf{f} \in \mathbf{R}^n, \lambda \in \mathbf{R} \tag{3}$$

These n equations of $n+1$ variables define in the $n+1$-dimensional space a one-dimensional curve $x(\lambda)$ passing through the operating point of the power system (x_0, λ_0). The indirect method is to start from (x_0, λ_0), and produce a series of solution points (x_i, λ_i) in a prescribed direction, determined by participating load and generation variations, until the 'nose' point is reached. A straightforward such method is to differentiate (3) with respect to λ:

$$\mathbf{f_x}(\mathbf{x},\lambda)\frac{d\mathbf{x}}{d\lambda} + \frac{\partial \mathbf{f}}{\partial \lambda} = \mathbf{0} \tag{4}$$

where,

$$\frac{\partial \mathbf{f}}{\partial \lambda} = \left[\frac{\partial f_1}{\partial \lambda}, \cdots, \frac{\partial f_n}{\partial \lambda}\right]^T \tag{5}$$

and then, solve (4) for $\frac{d\mathbf{x}}{d\lambda}$:

$$\frac{d\mathbf{x}}{d\lambda} = -\mathbf{f_x}^{-1}(\mathbf{x},\lambda)\frac{\partial \mathbf{f}}{\partial \lambda} \tag{6}$$

Integrating equation (6), one can get the solution curve $x(\lambda)$ on some interval $[\lambda_0, \lambda_1]$. The existence of a unique solution curve passing through (x_0, λ_0) is guaranteed by the implicit function theorem as long as the $\mathbf{f_x}(\mathbf{x},\lambda)$ remains invertible. However, this method fails when $\mathbf{f_x}(\mathbf{x},\lambda)$ becomes singular. Special indirect methods have been constructed to overcome this difficulty, such as continuation methods. There are many variations of continuation methods described in the literature. One of the most widely used is the predictor–corrector type. We will discuss this type of continuation method in some detail.

4. Eliminating Ill-conditioning

It is known that the set of power flow equations near its 'nose' point is ill-conditioned, making the Newton method diverge in the neighborhood of 'nose' points. From a numerical analysis viewpoint, this is due to the fact that, at the 'nose' point (i.e. the saddle-node bifurcation point), the two equilibrium points coalesce to form an equilibrium point, say x^* $(= x_s(\lambda^*) = x_1(\lambda^*))$. The Jacobian matrix evaluated at x^* has one zero eigenvalue, causing the set of power flow equations to be ill-conditioned.

There are several possible means to resolve the numerical difficulty arising from the ill-conditioning. One effective way, adopted in CPFLOW, is as follows: First, treat the parameter λ as another state variable

$$\mathbf{x_{n+1}} = \lambda.$$

Second, introduce the arclength s on the solution curve as a new parameter in the continuation process. This parameterization process gives

$$x = x(s), \quad \lambda = \lambda(s) = x_{n+1}.$$

The step-size along the arclength s yields the following constraint:

$$\sum_{i=1}^{n}\{(x_i - x_i(s))^2\} + (\lambda - \lambda(s))^2 - (\Delta s)^2 = 0$$

Third, solve the following n+1 equations for the n+1 unknowns x and λ

$$f(x, \lambda) = 0 \qquad (7)$$

$$\sum_{i=1}^{n}\{(x_i - x_i(s))^2\} + (\lambda - \lambda(s))^2 - (\Delta s)^2 = 0 \qquad (8)$$

It can be shown that the above set of augmented power flow equations is well-conditioned, even at the 'nose' point. CPFLOW solves these augmented power flow equations to obtain the solution curve passing through the 'nose' point without encountering the numerical difficulty of ill-conditioning.

5. Continuation Methods

Continuation methods, sometimes called curve tracing or path following, are useful tools to generate solution curves for general nonlinear algebraic equations with a varying parameter. Continuation methods have been applied successfully to a variety of engineering problems including electric power systems [7,8]. CPFLOW uses continuation methods to trace power system steady-state behavior due to load and generation variation.

The theory of continuation methods has been studied extensively and has its roots in Algebraic Topology and Differential Topology. Continuation methods have four basic elements:

- Parameterization
- Predictor
- Corrector
- Step-size control

Parameterization

Parameterization is a mathematical way of identifying each solution on the solution curve so that 'next' solution or 'previous' solution can be quantified. There are three different types of parameterizations:

1. physical parameterization using the controlling parameter λ, in which case the step length is $\Delta\lambda$,
2. local parameterization, which uses either the controlling parameter λ or any component of the state vector x; namely x_k to parameterize the solution curve. The step length in the local parameterization is $\Delta\lambda$ or Δx_k, and
3. arclength parameterization employing the arclength along the solution curve to perform parameterization, the step length in this case is Δs;

$$\Delta s = \{\sum_{i=1}^{n}(x_i - x_i(s))^2 + (\lambda - \lambda(s))^2\}^{0.5}$$

The so-called pseudo arclength parameterization uses different weighting factors (instead of an equal weighting factor) in the above equation.

While using the controlling parameter to parameterize the solution curve has physical significance, it encounters numerical difficulties in the vicinity of a saddle-node bifurcation point. In order to resolve this issue and to design an effective predictor, CPFLOW utilizes the arclength parameterization.

Predictor

The purpose of the predictor is to find an approximation point for the next solution. Suppose we are at the i-th step of the continuation process and the i-th solution (x^i, λ^i) of (3) has been found. The predictor attempts to find an approximation point for the next solution (x^{i+1}, λ^{i+1}). The quality of the approximation point by a predictor significantly affects the number of iterations required by a corrector in order to obtain an exact solution. A better approximation point yields a fewer number of iterations needed in a corrector to reach the solution. Several different predictors have been proposed in the literature of numerical analysis. They can be divided into two classes: (1) ODE based methods, which use the current solution and its derivatives to predict the next solution. The tangent method, a popular one as a predictor, is a first-order ODE-based method; (2) polynomial extrapolation based methods; which use only current and previous solutions to find an approximated solution. The secant method, a popular polynomial-based predictor, uses the current solution and the previous one to predict the next one. In order to achieve computational efficiency, CPFLOW employs the tangent method in the first phase of solution curve tracing and the secant method in the second phase.

Tangent method

The Tangent method calls for the calculation of the derivatives of $x_1, x_2, \cdots, x_n, x_{n+1}$ with respect to the arclength s

$$\frac{dx_1}{ds}, \ldots, \frac{dx_n}{ds}, \frac{dx_{n+1}}{ds}$$

To find these derivatives, differentiate both sides of equation (3) with respect to s:

$$0 = f_x \frac{dx}{ds} + f_{x_{n+1}} \frac{dx_{n+1}}{ds} \qquad (9)$$

Equation (9) is an implicit system of n linear algebraic equations in $n+1$ unknowns

$$\frac{dx_i}{ds}, \qquad i = 1, 2, ..., n, n+1 \qquad (10)$$

with the coefficients being the elements of the matrix:

$$Df = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_k} & \cdots & \frac{\partial f_1}{\partial x_{n+1}} \\ \frac{\partial f_2}{\partial x_1} & \cdots & & \cdots & & \cdots \\ \vdots & \vdots & & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_k} & \cdots & \frac{\partial f_n}{\partial x_{n+1}} \end{bmatrix} \qquad (11)$$

The following equation is required to make sure that s is the arclength on the curve.

$$(\frac{dx_1}{ds})^2 + \ldots + (\frac{dx_n}{ds})^2 + (\frac{dx_{n+1}}{ds})^2 = 1 \qquad (12)$$

Note that equations (9) and (12) form a set of $n+1$ equations in $n+1$ variables. Also, notice that (9) is a set of linear equations in its $n+1$ unknowns and (12) is nonlinear. A special method to solve this $n+1$ equations due to Kubicek [9] is as follows:

Suppose

$$dx_k/ds \neq 0 \tag{13}$$

for some k, $1 \leq k \leq n+1$, and let Df_k be the matrix of Df with the k-th column taken out, and suppose Df_k is not singular, then equation (9) can be solved for the unknowns

$$\frac{dx_1}{ds}, \; \cdots \; \frac{dx_{k-1}}{ds}, \; \frac{dx_{k+1}}{ds}, \; \cdots \; \frac{dx_{n+1}}{ds} \tag{14}$$

in the form of

$$\frac{dx_i}{ds} = \beta_i \frac{dx_k}{ds} \quad i \neq k, i = 1, \cdots, n+1 \tag{15}$$

The solution for the coefficients β_i can be obtained by applying Gaussian elimination to the matrix Df_k. In fact, if one performs the Gaussian elimination with pivoting to the full set matrix Df, the column index k is also found.

Substituting (15) into (12), one gets

$$\left(\frac{dx_k}{ds}\right)^2 = \left(1 + \sum_{i=1, i\neq k}^{n+1} {\beta_i}^2\right)^{-1} \tag{16}$$

Equations (15) and (16) constitute the explicit expression of the derivatives of the curve $\mathrm{x}(\lambda)$ passing through the j-th continuation point $(\mathrm{x}^j, \lambda^j)$ with respect to arclength s. A predictor step can be accomplished by integrating one step further in the prescribed direction with the step-size h:

$$\hat{\mathrm{x}}_j^{i+1} = \mathrm{x}_j^{i+1} + h \frac{dx_j}{ds}, \quad j = 1, \cdots, n+1 \tag{17}$$

In the context of computational efficiency, one has to keep in mind that the evaluation of the β_i's involves solving a set of linear algebraic equations which could be time-consuming. Thus it is advantagous to use numerical procedures which require fewer such evaluations. This consideration prompts the use of the secant method as a predictor after the tangent method produces two approximate points.

<u>Secant method</u>

The polynomial extrapolation methods are based on a polynomial of varying order that passes through the current solution and previous solutions (x^i, λ^i), (x^{i-1}, λ^{i-1}), $\cdots$, to provide an approximation point for the next solution (x^{i+1}, λ^{i+1}). A trivial predictor is the zero-order polynomial which uses the current solution as an approximation point for the next solution; i.e.

$$(\hat{x}^{i+1}, \hat{\lambda}^{i+1}) = (x^i, \lambda^i)$$

A slightly modified predictor based on the zero-order polynomial is

$$(\hat{x}^{i+1}, \hat{\lambda}^{i+1}) = (x^i, \lambda^{i+1})$$

A predictor, known as the secant predictor, uses a first-order polynomial (a straight line) passing through the current and previous solutions to predict the next solution;

$$(\hat{x}^{i+1}, \hat{\lambda}^{i+1}) = (x^i, \lambda^i) + h(x^i - x^{i-1}, \lambda^i - \lambda^{i-1}) \tag{18}$$

where h_i is an appropriate step-size. Predictors based on higher-order polynomial can be similarly derived. It has been experienced that lower-order predictors are more effective in the long run. The secant method is available in CPFLOW. It is interesting to note that the modified zero-order predictor is implicitly used in the repeated power flow approach.

In general, $(\hat{x}, \hat{\lambda})$ is not a solution of $\mathrm{f}(\mathrm{x}, \lambda) = 0$, rather it is an initial guess for the corrector iterations that will hopefully converge to a solution within the specified tolerance. The distance between (x^i, λ^i) and $(\hat{x}^{i+1}, \hat{\lambda}^{i+1})$ is called the *step length*. On the other hand, the measure of distance between (x^i, λ^i) and (x^{i+1}, λ^{i+1}) is given by the parameterization strategy, for example, arc-length.

<u>The Corrector</u>

After the predictor has produced an approximation $(\overline{\mathrm{x}}^{j+1}, \overline{\lambda}^{j+1})$ for the next solution $(\mathrm{x}^{j+1}, \lambda^{j+1})$, the error must be corrected before it accumulates. In principle, any effective numerical procedure for solving a set of nonlinear algebraic equations can be used for a corrector. Since a good predictor gives an approximation in a neighborhood of the next solution $(\mathrm{x}^{j+1}, \lambda^{j+1})$, a few iterations usually suffice for an appropriate corrector to achieve the needed accuracy.

The Newton method is chosen in CPFLOW as the corrector. This choice has an additional advantage; that is, the existing power flow computer packages based on the Newton method can be modified, with some effort, to serve as a corrector.

<u>Step Length Control</u>

One key element affecting the computational efficiency associated with a continuation method is the step-length control. It is safe to choose a constant, small step length in any continuation method. However, this constant step length may often lead to inefficient computation, such as too many steps through the 'flat' part of the solution curve. Similarly, an inadequately large step length can cause the predicted point (produced by the predictor) to lie far away from the (true) solution point, and as a result, the corrector needs many iterations to converge. In the extreme case, the corrector may diverge. Ideally, the step length should be adapted to the shape of the solution curve to be traced: a larger step length should be used in the 'flat' part of the solution curve and a smaller step-length in the 'curly' part (part with high degree of curvature) of the solution curve. Of course, the shape of the solution curve is unknown beforehand, making the task of designing an effective step length control difficult. Thus, good step length controls are usually custom designed for specific applications. Despite this, some general considerations may be implemented in the continuation procedure in order to improve its performance.

One strategy for step length control is to set up an upper limit $h_{max,i}$ for each variable x_i. The actual step length h along the arclength s is thus chosen such that

$$h\frac{dx_i}{ds} \leq h_{max,i}, \quad i = 1, \; \cdots, \; n+1 \tag{19}$$

The motivation for such an implementation is that the curve $\mathrm{x}(\lambda)$ under consideration may be "flat" with respect to some x_i, while turning sharply with respect to some other x_j. By assigning $h_{max,i}$ accordingly, that is, giving a larger $h_{max,i}$ to those variable along which the curve is "flat" and smaller $h_{max,i}$ otherwise, we can make the continuation process trace quickly through the "flat" portion of the curve and yet keep small steps through the "curly" portion. This in turn will yield a better approximation from the predictor, thus faster convergence for the corrector. The success of this step length control method depends greatly on the proper value of $h_{max,i}$, which requires prior knowledge of the problem under consideration. In the case of power system studies, experience provides good guidance. For example, the $h_{max,i}$

corresponding to a bus voltage would be given a small value since the whole range for this variable is about $[0, 1.2]$, while the $h_{max,i}$ corresponding to the reactive parameter λ should be assigned a larger value.

Another simple method is to observe the number of iterations taken at each continuation step. By setting a desired target number of iterations, the method compares the actual number of iterations to the target. If the actual number is smaller, then the next step length can be a little larger than the previous one. On the flip side, if the actual number is greater, then the next step length should be a little smaller than the previous one. However, this method fails to achieve the desired results if any control device (switchable shunt, ultc transformers, etc.) is forced out of its normal operating region. For example, if an ULTC transformer model adjusts a tap setting to bring the controlled bus voltage within its specified tolerance, then it will take a few extra iterations for the Newton method to converge. These extra iterations would shorten the following predictor step, causing the predictor to take shorter steps, hence making it less efficient.

CPFLOW computes the arclength in the state space, which automatically forces the predictor to take large steps on the 'flat' part of the solution curve and small steps on the 'curly' part. The efficiency of this approach will be demonstrated later via numerical studies.

It should be pointed out that step length controls must be designed coherently with the predictor, corrector and the problem under consideration. In the design of a predictor-corrector continuation method there is a delicate balance between speed and robustness. The key issues here are (i) robustness - whether a predicted point produced by a low-order predictor with a step-size chosen to be as large as possible lies inside the domain of convergence of the corrector; noting that the number of iterations in the corrector process may be large, and (ii) speed - the amount of total work required to solve the given problem to a specified accuracy.

6. Solution Algorithm

CPFLOW uses a predictor-corrector continuation method to trace solution curves. There are two types of predictors employed in CPFLOW. In phase one of CPFLOW, the Tangent method serves as a predictor. After at least two solution points have been obtained, the predictor can be switched to the Secant method which requires the knowledge of the current and previous solutions (phase two). A step-by-step exposition of the solution algorithm used in CPFLOW is summarized in Figure 1.

7. Numerical Studies

CPFLOW has been applied to several large-scale power systems including the following:

Network Components
Buses: 3493, Swing bus:1, Generators: 844, Loads: 2565, fixed shunts: 957, switchable shunts: 125, Lines: 5953, Fixed Transf: 654, Fixed phase shifter:1, ULTC Transf: 82, ULTC phase shifter: 8, Areas: 30, Zones: 23.

Due to space limitation, one numerical simulation examining reactive power increases is presented. The reactive power load is increased about 600 % at the four load buses in Area 6, which has a total of eight buses, 2 generators, and an ULTC transformer. Bus 635 is one of the load buses in Area 6 and it is also the controlling bus for the ULTC transformer

Step 1: Input power system data
Bus data, generator data, load data, fixed and/or switchable shunt data, branch data (including line data, fixed transformer data, ULTC transformer data, fixed and/or variable series compensator data, static tap-changer/phase-shifter data), area interchange data.

Step 2: Initialization
(i) Run a power flow study to make sure the initial point to be used in the continuation process lies on the solution curve.
(ii) Build up the current load demand pattern (P_d^0, Q_d^0), and real power generation pattern (excluding the slack bus) (P_g^0).

Step 3: Input the buses at which loads and/or real power generations are to be varied, and also input the quantity of variations.
(i) Build up the future load demand pattern (P_d^1, Q_d^1), and real power generation pattern (P_g^1).
(ii) Parameterize the load demand pattern $(P_d(\lambda), Q_d(\lambda))$, and real power generation pattern $(P_g(\lambda))$ so that (see Fig. 2)

$$(P_d(0), Q_d(0), P_g(0)) = (P_d^0, Q_d^0, P_g^0)$$

$$(P_d(1), Q_d(1), P_g(1)) = (P_d^1, Q_d^1, P_g^1)$$

Then form the vector sum of the bus injection power pattern.

Step 4: Predictor (phase-one)
(i) Calculate the derivatives $(dx_1/ds, \cdots, dx_n/ds, dx_{n+1}/ds)$ via β's.
(ii) Choose an appropriate step-length.
(iii) Obtain the predicted point using equation (17).

Step 5: Predictor (phase-two)
After at least two solution points have been obtained, the predictor can be switched to the secant method which requires the knowledge of the current and previous solutions.
(i) choose an appropriate step-length.
(ii) obtain the approximation point using equation (18).

Step 6: Corrector (phase-one)
Solve the set of augmented power flow equations (7) and (8) using the Newton method starting from the approximation point obtained in the Predictor (phase-one).

Step 7: Corrector (phase-two)
Solve the set of augmented power flow equations (7) and (8) using the Newton method starting from the approximation point obtained in the Predictor (phase-two).

Step 8: Stop criterion
If further tracing of solution curve is of no interest, stop here. Otherwise, go to Predictor step to calculate the next point.

Figure 1: Solution Algorithm Structure

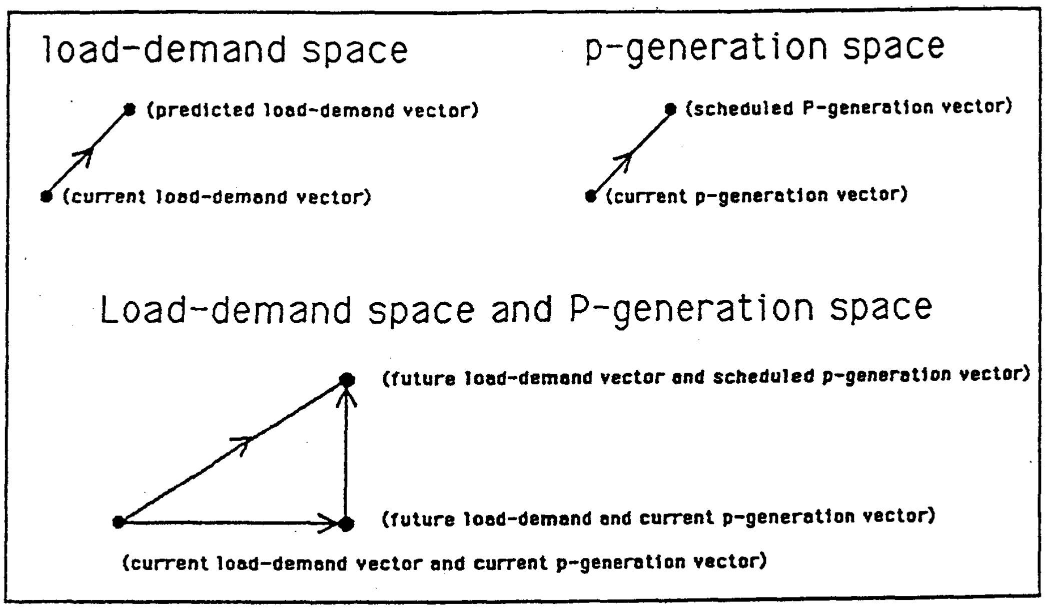

Figure 2: Parameterize both the load demand pattern and real power generation pattern

joining Bus 635 and Bus 636. All controls are active in this simulation, which causes the initially flat voltage magnitude at Bus 635. Then at $\lambda = 1.4$ the ULTC tap ratio reaches its upper limit and the voltage magnitude begins to drop.

Figure 3 shows the voltage magnitude at Bus 635 decreasing as the reactive power increases. Figure 4 shows the voltage angle remaining nearly constant as the reactive power increases. Then, after the saddle node bifurcation point the angle rapidly decreases. Figure ?? and Figure ?? both demonstrate the two step predictor–corrector continuation method. Along the stable equilibrium branch, the predictor step length is fairly large, but as the saddle node bifurcation point is approached, the parameter step length shrinks. These two figures illustrate an important point, that the step length varies in the parameter space, attempting to adapt itself to the shape of the solution curve.

Unfortunately, statistically significant timing data has not yet been compiled for CPFLOW. However, the execution time for the numerical study above will give a crude estimate of the relative improvement of CPFLOW over the normal repeated power flow solution approach. The numerical simulation for the 3500 bus case required 15 minutes on a SUN SPARCstation 2. The simulation took 69 continuation steps with approximately 5 seconds spent in the predictor routines and 6 seconds spent in the corrector routines per step. This left approximately 2 minutes for loading data and building the basic power flow data structures.

8. Conclusions

A detailed description of the implementation regarding the predictor, corrector, step-size control and parameterizations employed in CPFLOW has been presented. A two-stage solution algorithm has been developed and implemented in CPFLOW. CPFLOW overcomes numerical ill-conditioning at the 'nose' point by solving an augmented set of power flow equations. CPFLOW achieves its computational speed via an effective predictor-corrector, adaptive step-length selection and efficient I/O operations. CPFLOW is capable of

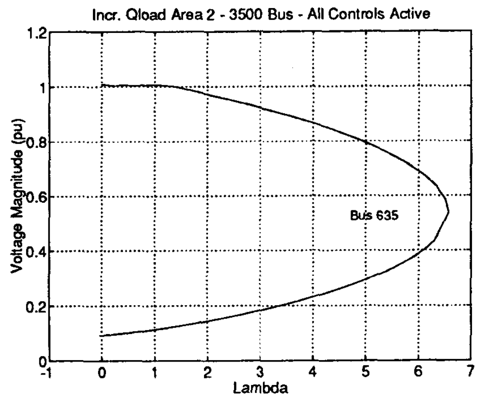

Figure 3: Voltage magnitude at bus 635 due to reactive power increase in Area 6.

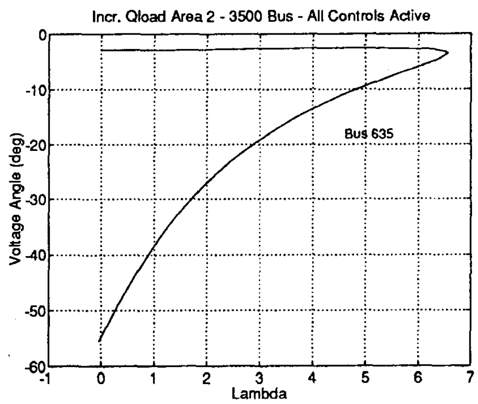

Figure 4: Voltage angle at bus 635 due to reactive power increase in Area 6.

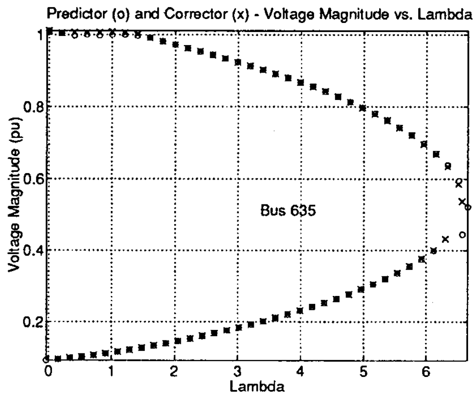

Figure 5: The performance of predictor and corrector during the continuation process for voltage magnitude

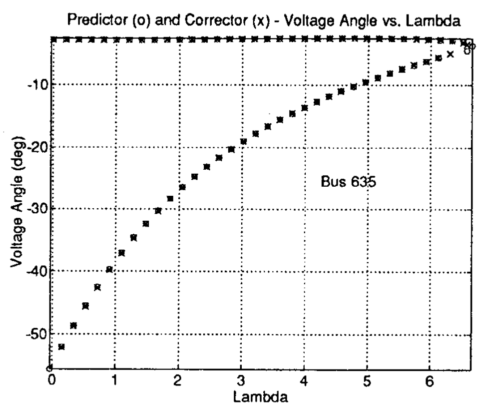

Figure 6: The performance of predictor and corrector during the continuation process for voltage angle

handling both load variations and real power generation variations through a proper parameterization scheme. While this scheme only allows for the parameterization of the constant P-Q portion of total bus load, CPFLOW can model load as any function of voltage. Numerical studies on a 3500-bus power system with a comprehensive set of operational limits and controls are discussed.

We remark on future possible work of improving the computational speed and reliability of CPFLOW. In the design of a predictor-corrector continuation method there is a delicate balance between speed and robustness. The key issue is whether a predictor-corrector continuation method can be designed wherein the predicted point produced by the predictor remains 'close' to the solution curve for robustness and yet the overall amount of computational effort required to solve a given problem is minimized. We suspect that the design is dependent on both problem characteristics and size. Indeed, we have found that we often need to make several runs of CPFLOW at different error tolerances when we are trying to achieve greater computational efficiency. Furthermore, from the viewpoints of both theoretical analysis and practical implementation of predictor-corrector continuation methods, the trade-offs between robustness and minimum computational effort remain open.

9. Acknowledgements

The authors wish to thank the useful discussions they had with Mr. Dave Takach of Union Electric, Dr. Weimin Ma (presently with ESCA) and Dr. Jianzhong Tong of Cornell University, which have resulted in several improvements in CPFLOW. Mr. Takach's constructive comments on the paper are also greatly appreciated.

References

[1] R. Seydel, *From Equilibrium to Chaos: Practical Bifurcation and Stability Analysis*, New York: Elsevier, 1988.

[2] EPRI Final Report of Interactive Power Flow (IPFLOW), April 1992.

[3] H.D. Chiang, W. Ma, R.J. Thomas, and J.S. Thorp, "A Tool for Analyzing Voltage Collapse in Electric Power Systems," *Proceedings of the* 10^{th} *Power Systems Computation Conference*, Graz, Austria, August, 1990.

[4] K. Iba, H. Suzuki, M. Egawa, T. Watanabe, "Calculation of the Critical Loading Condition with Nose Curve Using Homotopy Continuation Method," *IEEE Trans. on Power Systems*, Vol. 6, No. 2, May 1991, pp. 584-593.

[5] V.A. Ajjarapu and C. Christy, 'The Continuation Power Flow: A Tool for Steady State Voltage Stability Analysis', *IEEE Trans. on Power Systems*, Vol. 7, No. 1, Feb. 1992, pp.416-423.

[6] C.A. Cañizares and F.L. Alvarado, 'Point of Collapse and Continuation Methods for Large AC/DC Systems', *IEEE Trans. on Power Systems*, Vol. 8, No. 1, Feb. 1993, pp. 1-8.

[7] M. Huneault, A. Fahmideh-Vojdani, M. Juman, R. Calderon, and F.G. Galiana, 'The Continuation Method in Power System Optimization: Applications to Economy Security Functions',*IEEE Trans. on PAS*, Vol.104, No. 1, 1985, pp. 114-124.

[8] M. Huneault and F.G. Galiana, 'An Investigation of the Solution to the Optimal Power Flow Problem Incorporating Continuation Methods', *IEEE Trans. on Power Systems*, Vol. 5, No. 1, Feb. 1990, pp.103-110. pp.416-423.

[9] M. Kubiček, M. Marek, *Computational Methods in Bifurcation Theory and Dissipative Structure*, Springer Series in Computational Physics, Springer-Verlag, 1983

Hsiao-Dong Chiang received the Ph.D. degree in electrical engineering and computer sciences from the University of California at Berkeley. He is currently an Associate Professor of Electrical Engineering at Cornell University, Ithaca, New York. He was a recipient of the Engineering Research Initiation Award (1988) and of the Presidential Young Investigator Award (1989) from the National Science Foundation. In 1990 he was selected by a Cornell Merrill Presidential Scholar as the faculty member who had the most positive influence on that student's education at Cornell. His research interests include power systems, nonlinear systems, optimization techniques and neural networks. He was an Associate Editor of the IEEE Transactions on Circuits and Systems (1990-1991).

Alexander J. Flueck received the B.S. degree (May 1991) and the M.Eng. degree (Aug. 1992) in electrical engineering from Cornell University. He is currently working on a Ph.D. at Cornell with Professor Chiang. His research interests include power system analysis, control system theory and nonlinear system theory.

Kirit S. Shah received the Ph.D. degree in electrical engineering from the University of Missouri, Rolla., in 1976. He is currently Supervising Engineer-Transmission and Interconnections/Metro Substransmission and Bulk Substations for the System Planning Department of Union Electric. He is active in various Mid-America Interconnected Network(MAIN) Committees. He has been involved in several EPRI research projects and is a member of the EPRI Power System Planning and Operations Program (PSPO) Task Force.

Dr. Neal Balu is the program manager of the Power System Planning and Operations Program of EPRI's Electrical Systems Division, where he is responsible for formulation and developing research strategies for the program. Before joining EPRI in 1979, he was a section manager in the System Planning Department of Southern Company Services in Birmingham, Alabama. Dr. Balu holds a PhD in electrical engineering from the University of Alabama and an MBA from Santa Clara University, Santa Clara, CA and is a senior member of the IEEE.

Discussion

Claudio A. Cañizares* and Fernando L. Alvarado** (*University of Waterloo, **University of Wisconsin–Madison): This interesting paper presents another practical implementation of the continuation method for tracing bifurcation diagrams of power systems. There are several differences between this particular implementation and the one presented by the discussers in [6]. Hence, we would appreciate the comments of the authors on the following points:

1. The authors propose using an arclength parameterization technique to guarantee having a nonsingular Jacobian matrix for the predictor step, when the system is close to the saddle-node bifurcation point. Is there any particular advantage of this technique over the much simpler one used in [6]?

   The technique in reference [6] is based on an interchange between the voltage variable that changes the most and the parameter λ. The method also guarantees a nonsingular Jacobian at the bifurcation point, as formally proven in [A1]. However, this method generally requires less computational effort as a result of better sparsity properties. Furthermore, as mentioned in [6], the discussers have observed that in all practical applications, regardless of system size and complexity, there is no rel need for the parameterization step when step size control is used. This is possible because the system Jacobian only becomes singular when very close to the bifurcation, as shown by the behavior of the system eigenvalues depicted in Fig. 1 for a small system, which agrees with a similar behavior of singular values in larger systems depicted in [A2]. The latter explains why engineers in the power industry have been able to get rather close to a bifurcation point by successively solving the power flow equations [A2]. Based on this fact, the discussers have proposed using a mixed technique to be implemented in a EPRI program (VSTAB), which consists on using successive power flows with a constant parameter change until the technique diverges, and then switch to a continuation method to trace the bifurcation diagram from the last solution obtained with the standard power flow; this significantly reduces the computational time needed for tracing bifurcation diagrams.

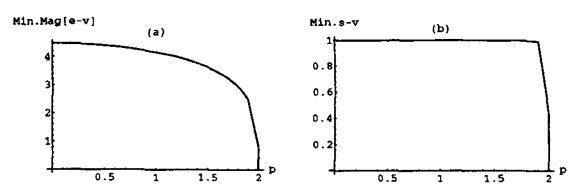

Fig. 1: *(a) Minimum magnitude of the eigenvalues and (b) minimum singular value as a function of the parameter P, for a generator-line-load system.*

2. The authors use either a tangent or a secant method in the predictor step, with the advantage of the secant method being faster in yielding the new variables values for the corrector step. However, the tangent method has the benefit of producing an approximation to the left eigenvector **v** at the saddle-node bifurcation point, since the tangent vector smoothly converges to **v** [A1]. This eigenvector contains significant information for sensitivity analysis of the system at or close to the bifurcation point. The discussers have experienced convergence difficulties with the secant method when limits are encountered (particularly Q-limits), due to sudden changes in the direction of the bifurcation diagram that cannot be accurately predicted by the secant method. Have the authors experimented with a mixed approach, consisting of using the secant method until limits are encountered or when close to bifurcation, to switch then to a tangent vector approach?

3. In [6] the discussers commented on the difficulties of handling limits and the problems that these present from the computational performance point of view, due to the need of having to refactorize matrices when some limits are encountered, and particularly since one cannot apply multiple limits at once without running the risk of tracing the wrong bifurcation diagram. Furthermore, HVDC links pose an additional problem, due to the large number of limits and control strategies that are used to operate these systems, as discussed in detail in [6]. Hence, the techniques have been used by the authors to handle limits without compromising computational efficiency, and particularly, what switching criteria have been used to correctly handle HVDC links?

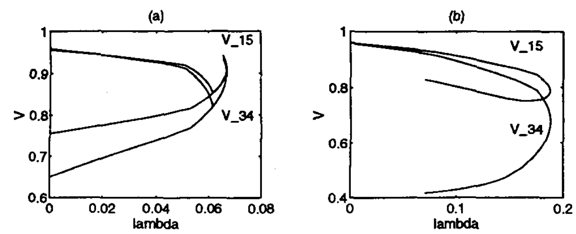

Fig. 2: *Bifurcation diagrams of the National Interconnected Ecuadorian system for (a) PQ and (b) mixed load models.*

4. Finally, the authors assume a constant PQ variation of the system loads, which has been a typical approach in bifurcation studies of power systems. For these load models, the saddle-node bifurcation point is equivalent to the maximum available power of the transmission system. However, other steady-state load models such as constant impedance loads or voltage dependent load models, yield significantly different bifurcation diagrams and locations of saddle-node bifurcations [A3, A4], as shown in the bifurcation plots of Fig. 2 for a real system. In Fig. 2(a), the system reaches a Q-limit instability point of the type discussed in [6] and [A5], for a relatively small value of the parameter λ, whereas in Fig. 2(b) the "loadability margin", or maximum λ, has significantly increased and no Q-limit instabilities are encountered. Have the authors considered including different static load models in CPFLOW?

[A1] C. A. Cañizares, "Voltage Collapse and Transient Energy Function Analyses of AC/DC Systems," *Ph. D. Thesis*, University of Wisconsin–Madison, September 1991.

[A2] Y. Mansour, ed., *Suggested Techniques for Voltage Stability Analysis*, IEEE/PES, 93TH0620-5PWR, 1993.

[A3] C. A. Cañizares, W. L. Long, F. L. Alvarado, C. L. DeMarco, "Techniques for Detecting Proximity to Voltage Collapse in AC/DC Systems," *Proceedings III SEPOPE*, Bello Horizonte, Brazil, May 1993, paper IP-18.

[A4] F. L. Alvarado, "Voltage Stability, Bifurcation Parameters and Continuation Methods," *Proceedings IV SEPOPE*, Foz du Iguaçu, Brazil, May 1994.

[A5] I. Dobson and L. Lu, "Voltage Collapse Precipitated by the Immediate Change in Stability When Generator Reactive Power Limits are Encountered," *IEEE Trans. Circuits and Syst.–I*, Vol. 39, No. 9, September 1992, pp. 762–766.

Manuscript received February 17, 1994.

T. WU, R. FISCHL and S. WUNDERLICH (Drexel University, Philadelphia, PA): We congratulate the authors for successfully applying the continuation method [9] to solve the load flow feasibility problem of a power system model that contains reactive power limits, ULTC transformers, ULTC phase shifters and VAR compensators. This problem is very difficult since when the system operating conditions hit a limit the load flow equations change. For example, when a generator hits its VAR limits, its model changes from a PV bus to a PQ bus. This implies an increase in the number of equations in (3) as well as the dimensionality of the solution space. This can be stated mathematically as follows:

$$\mathbf{f}_a(\mathbf{x}_a, \lambda) = \mathbf{0} \tag{D1}$$

$$a = \{a \in \mathbf{I}^+ \mid \mathbf{w}_a(\mathbf{x}_a, \lambda) \leq \mathbf{0}\} \tag{D2}$$

where a is a parameter that identifies the process of hitting a constraint limit; and $\mathbf{f}_a:\mathbf{R}^{n(a)+1} \to \mathbf{R}^{n(a)}$, $\mathbf{x}_a \in \mathbf{R}^{n(a)}$, $\lambda \in \mathbf{R}$, $\mathbf{w}_a:\mathbf{R}^{n(a)+1} \to \mathbf{R}^m$. Note, (D1) represents a sequence of power flow equations whose dimension is parameterized by a, namely $n(a)$ The parameter a (i.e., the sequence of power flow equations) depends on the parameter λ. Our questions are:

(1) Since a continuation method is usually applied to a smooth function, we would like to know the theoretical basis that justifies the use of the continuation method globally for a sequence of equations whose dimension changes as a function of λ?

(2) From our numerical experiences, the predicted point based on a power flow equation before reaching a limit may be out of region of convergence for the power flow equation after reaching the limit especially when being close to the turning point. Has CPFLOW some mechanism which takes care of this problem?

Manuscript received February 22, 1994.

Hsiao-Dong Chiang and Alexander J. Flueck:

The authors would like to thank the discussers for their interest in our paper and their valuable comments. We will attempt to address individually their respective concerns in the following closure.

To Dr. Cañizares and Dr. Alvarado:

(1) While it is true that the use of arclength parameterization will eliminate the Jacobian singularity at a saddle node bifurcation point, this is not the most important issue in continuation method design and implementation. Rather, the condition of the Jacobian throughout the entire continuation trace is the most important consideration. Futhermore, a nonsingular Jacobian is not necessary in the predictor step, since a secant predictor may be employed, though it is crucial in the corrector step.

The authors believe that arclength parameterization is generally more robust than local parameterization. A tradeoff exists between computational burden and robustness. While the arclength parameterization appends a dense row to the Jacobian, it does not affect the sparsity properties of the original Jacobian. The augmented Jacobian has only the fill-in elements of the original Jacobian, even though it has more non-zero elements.

The authors believe that the increased robustness of arclength parameterization, combined with an acurate and efficient predictor will allow the continuation method to take larger steps than in the case of local parameterization and flat prediction.

(2) The authors agree that the tangent predictor at the saddle node bifurcation point contains valuable information as the right eigenvector corresponding to the zero eigenvalue. However, it is possible to use a secant predictor to trace the entire continuation curve and only upon detection of a saddle node bifurcation to compute the right eigenvector via the tangent predictor.

A tradeoff exists between the computational burden of predicting a solution and the accuracy of the predicted solution. In general, the tangent is more accurate and requires fewer corrector iterations to find a tolerable solution at the expense of Jacobian factorizations. On the other hand, the relatively inexpensive secant is based on old information which can lead to an increased number of corrector iterations. These extra corrector iterations may involve extra undesirable Jacobian factorizations or they may not, since the Jacobian is not necessarily evaluated in each corrector iteration. Note that the tangent predictor is only more accurate than the secant in predicting a solution to the current set of equations, i.e., the tangent predictor cannot forsee control limit effects any better than the secant.

A priori, it is difficult to tell how the tradeoff will affect performance of a continuation method. In some situations, the secant predictor may yield an overall more efficient continuation method than the tangent predictor as explained in the following scenario. The authors have been investigating interarea real power transfer capabilities of the 3493 bus case described in the paper. In this continuation study, many control limits are reached (e.g., generator reactive power limits, ULTC voltage limits, switchable shunt voltage limits, area interchange schedules, etc.). Although the inaccuracies of the secant method caused the corrector to take an extra iteration on some steps, it traced the entire curve of stationary solutions 8% faster than the tangent method.

(3) The robustness of arclength parameterization can usually handle control violations in tracing continuation curves. The physical parameter λ is not fixed, as in the successive power flow approach, allowing the corrector some flexibility in which to search for solutions to the constrained power flow equations. In the extreme, CPFLOW may not converge to a solution within the user specified mismatch tolerance and maximum number of iterations. In this case, the previous solution is restored and a user specified stepsize multiplier shrinks the desired stepsize and CPFLOW attempts a new continuation step.

HVDC links are modeled by CPFLOW in the same manner as they are modeled in the EPRI program IPFLOW, developed by Ontario Hydro.

(4) While CPFLOW is currently only able to parameterize the PQ loads at each bus, it can model all other loads in the network as any type of static voltage dependent load, as in IPFLOW.

At the same time, the authors would like to point out that the discussers' figures (2a, 2b) do not clearly illustrate the effects of different types of load models on the location of the saddle node bifurcation point. In Figure 2a, it is difficult to see where the curves turn around. There appears to be a jump discontinuity in the voltage magnitude plots for buses 15 and 34. Furthermore, it is not known how the load was varied nor whether λ represents constant PQ load or some type of voltage dependent load.

The authors agree that different load models might affect the location of the saddle node bifurcation point because the underlying model changes, although the effects have yet to be fully documented.

To Dr. Fischl, Mr. Wu and Mr. Wunderlich:

(1) At present the authors do not have a rigorous theoretical justification for the use of continuation methods on sequences of equations. Instead, we would like to offer an informal argument for the uniqueness of the continuation curve of solutions. First, suppose there exists a solution (x^1, λ^1) to the augmented system $(f^a(x,\lambda), p^a(x,\lambda))$. The curve of solutions passing through (x^1, λ^1) is unique in some neighborhood, as guaranteed by the Implicit Function Theorem, if the augmented Jacobian is nonsingular at (x^1, λ^1). Now suppose that the predicted point $(\hat{x}^2, \hat{\lambda}^2)$ violates a control constraint. When the corrector searches for a solution (x^2, λ^2), it may solve a different set of equations $(f^b(x,\lambda), p^b(x,\lambda))$ rather than the original set. However, the Implicit Function Theorem will again guarantee that the curve of solutions through (x^2, λ^2) is unique in some neighborhood when the Jacobian is nonsingular.

We now describe a simple three bus power system with one load (Bus 3) that has an increasing real power demand. The slack generator (Bus 1) has a very large reactive power reserve and its terminal voltage always remains at the specified setpoint. However, the other generator (Bus 2), whose voltage magnitude is shown in Figure 1, has a relatively limited capability to generate reactive power.

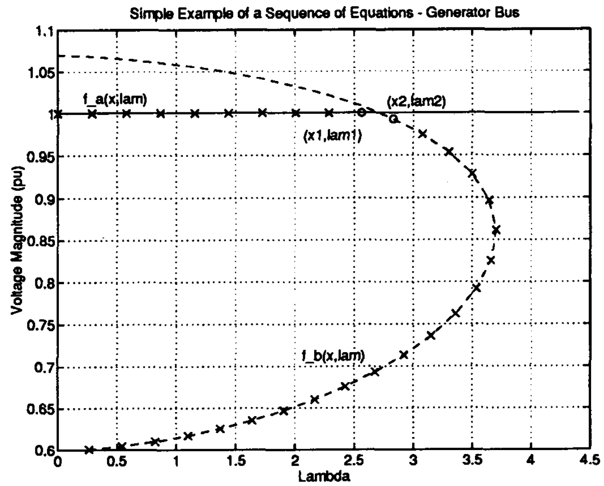

Figure 1: A Simple Sequence of Power Flow Equations Involving a Generator (Bus 2) Switching from a PV Bus to a PQ Bus

In Figure 1, each curve (the solid curve described by equation f^a and the dashed curve given by f^b) has been computed by CPFLOW. The $\times$ and o points are corrector solutions to the power flow equations and were computed by CPFLOW on yet a third run. The base case solution (step 0) to this simple power system is shown at $\lambda = 0.0$. We have chosen to represent continuation solutions 9 and 10 by circles to illustrate the transition from equation f^a to equation f^b which occurs between points (x^1, λ^1) and (x^2, λ^2). In equation f^a, the generator (Bus 2) reactive power limit is ignored, hence the terminal voltage remains fixed and the exact saddle node bifurcation value is $\lambda^* = 4.39799$. In equation f^b, the reactive power output is fixed at its maximum value, hence the voltage magnitude varies and the exact SNB value is $\lambda^* = 3.70132$. When tracing the stationary solution to this simple system as the real power demand at Bus 3 is varied, CPFLOW encounters the Bus 2 generator reactive power violation. At this point, the bus is switched from PV to PQ and the reactive power equation associated with Bus 2 enters the set of equations as illustrated by the $\times$ and o points.

Sometimes, this change from one set of equations to another, will force the corrector to take extra iterations in solving the power flow. Yet, given a sequence of control limit violations, CPFLOW will trace a unique continuation curve. However, if the sequence is reordered, then it is difficult to say how near or how far the two curves will be in relation to each other.

(2) The robustness of the arclength parameterization scheme generally allows CPFLOW to handle control lim-

its and their associated changes to the convergence regions. However, in the extreme, CPFLOW contains a reset state routine that will reload the previous solution, rescale the desired stepsize according to user specified parameters, and attempt to calculate a new solution to the constrained power flow equations.

In closing, the authors would again like to thank the discussers for taking the time to comment on our research.

Manuscript received November 16, 1994.