

# A JACOBIAN SMOOTHING NONLINEAR COMPLEMENTARITY METHOD FOR SOLVING NONLINEAR OPTIMAL POWER FLOWS

Geraldo L. Torres

Dept. de Engenharia Elétrica e Sistemas de Potência  
Universidade Federal de Pernambuco, Brazil  
gltorres@ieee.org

Victor H. Quintana

Dept. of Electrical and Computer Engineering  
University of Waterloo, Canada  
quintana@ieee.org

**Abstract -** The paper presents a Jacobian smoothing nonlinear complementarity method for solving large nonlinear optimal power flow (OPF) problems. By using (smooth or non-smooth) nonlinear complementarity (NC) functions we reformulate the OPF problem as a (smooth or non-smooth) nonlinear system of equations, which, in turn, can be solved by well-known methods. The Jacobian smoothing method computes the search directions from linear systems that are formed with the right-hand side from a non-smooth reformulation and the smooth matrix from a smooth reformulation. We test the computational efficiency of the proposed OPF algorithm on test systems of up to 2098 buses.

**Keywords -** Optimal Power Flow, Jacobian Smoothing, Nonlinear Complementarity Method.

## 1 INTRODUCTION

THE paper presents a Jacobian smoothing nonlinear complementarity method for solving large nonlinear optimal power flow (OPF) problems. Besides the classical active loss minimization problem, two other OPF variants are solved: maximization of the system loadability, and minimization of load shedding to restore feasibility of the system operation. The nonlinear OPF will be written as

$$\begin{aligned} & \text{Minimize} && f(x, \sigma) \\ & \text{subject to} && g(x, \sigma) = 0, \\ & && \underline{h} \leq h(x, \sigma) \leq \bar{h}, \end{aligned} \quad (1)$$

where  $x$  is an  $n$ -vector of decision variables (voltage magnitudes and angles,  $(V, \theta)$ , transformers tap ratios,  $t$ , shunt susceptances,  $b^{\text{sh}}$ , etc);  $\sigma$  is the system loading factor;  $f$  is a scalar function that represents the system operation optimization goal;  $g$  is an  $m$ -vector function with power balance equations and other hard constraints; and  $h$  is a  $p$ -vector of functional and simple bound variables, with lower bound  $\underline{h}$  and upper bound  $\bar{h}$ , that correspond to physical and operating limits on the system.

For maximum loadability computation  $f(x, \sigma) = -\sigma$  and the active and reactive power balance equations, for the buses indicated for load increase, are expressed as

$$\begin{aligned} P_i(V, \theta, t) - P_i^G + (1 + \sigma)P_i^L &= 0, \\ Q_i(V, \theta, t) - Q_i^G + (1 + \sigma)Q_i^L &= 0, \end{aligned} \quad (2)$$

where  $P_i$  and  $Q_i$  are net power injections,  $P_i^G$  and  $Q_i^G$  are power generations, and  $P_i^L$  and  $Q_i^L$  are power loads, all at the bus  $i$ . The  $\sigma$  variable represents the loading factor that simulates load changes that drive the system to collapse.

Since several operational limits are considered in the problem formulation (bounds on voltage magnitudes, on transformers tap ratios, on reactive power outputs of generators, on shunt reactive power compensations, and on line flows), a solution to this problem is not a collapse point. Instead, it is the maximum loading the system can sustain while satisfying the operational limits specified. Notice that we consider constant power changes only, and the load increase holds the load power factor. Notice also that a single parameter  $\sigma$  is used in all the buses indicated for load increase.

For minimum load shedding computation  $f(x, \sigma) = \sigma$  and the active and reactive power balance equations, for the buses indicated for load shedding, are expressed as

$$\begin{aligned} P_i(V, \theta, t) - P_i^G + (1 - \sigma)P_i^L &= 0, \\ Q_i(V, \theta, t) - Q_i^G + (1 - \sigma)Q_i^L &= 0. \end{aligned} \quad (3)$$

The base condition here is a system that cannot satisfy the operational limits without shedding some load, or even the extreme condition of an insolvable power flow. See [1, 2, 3] for further concerns on these OPF formulations.

For compactness of the presentation, we will consider the  $\sigma$  variable hereafter as the last entry of the  $x$  vector. If  $x_*$  is a local minimizer to (1), then there exist vectors of Lagrange multipliers, say,  $(\lambda_*, \pi_*, v_*)$ , that satisfy the Karush-Kuhn-Tucker (KKT) optimality conditions [4]:

$$-g(x) = 0, \quad (4)$$

$$s + z - \bar{h} + \underline{h} = 0, \quad (5)$$

$$h(x) + z - \bar{h} = 0, \quad (6)$$

$$\nabla f(x) - J_g(x)^T \lambda + J_h(x)^T v = 0, \quad (7)$$

$$S\pi = 0, \quad s \geq 0, \quad \pi \geq 0, \quad (8)$$

$$Z\hat{v} = 0, \quad z \geq 0, \quad \hat{v} \geq 0, \quad (9)$$

where  $s$  and  $z$  are slack vectors that transform the inequalities in (1) into the equalities (5) and (6);  $S$  and  $Z$  are diagonal matrices with  $S_{ii} = s_i$  and  $Z_{ii} = z_i$ ;  $\nabla f$  is the gradient of  $f$ ;  $J_g$  is the Jacobian of  $g$ ;  $J_h$  is the Jacobian of  $h$ ; and  $\hat{v} = v + \pi$ .

As explained in [4], the major difficulty in solving (4) to (9) is mainly related to the complementarity conditions (8) and (9). A ‘‘root-finding Newton’s method’’ applied to (4) to (9) cannot automatically assure  $(s, z, \pi, \hat{v}) \geq 0$ , and the numerical solution of  $S\pi = 0$  and  $Z\hat{v} = 0$  is intricate. For instance, the Newton equation for  $s_i \pi_i = 0$  is

$$s_i^k \Delta \pi_i + \pi_i^k \Delta s_i = -s_i^k \pi_i^k. \quad (10)$$

If a variable is zero, say,  $\pi_i^k = 0$ , then the Newton equation becomes  $s_i^k \Delta \pi_i = 0$ , leading to a zero update,  $\Delta \pi_i = 0$ . Consequently,  $\pi_i^k$  will remain zero all the time once it becomes zero, which is fatal because the algorithm will never be able to recover from such a situation.

A class of primal-dual interior-point (IP) methods, based on the logarithmic barrier function approach, avoid these numerical difficulties by considering the perturbed KKT equations

$$-g(x) = 0, \quad (11)$$

$$s + z - \bar{h} + \underline{h} = 0, \quad (12)$$

$$h(x) + z - \bar{h} = 0, \quad (13)$$

$$\nabla f(x) - J_g(x)^T \lambda + J_h(x)^T v = 0, \quad (14)$$

$$S\pi = \mu_k e, \quad (15)$$

$$Z\hat{v} = \mu_k e, \quad (16)$$

where  $\mu_k > 0$  is a barrier parameter that is forced to go to 0 as  $k \rightarrow \infty$ , and  $e = (1, 1, \dots, 1)^T$  is a vector of ones of appropriate dimension. Primal-dual IP methods do not require that (11) to (14) be satisfied at every iterate, but they require that  $(s, z, \pi, \hat{v}) > 0$  be verified at every point.

A new nonlinear complementarity (NC) approach has recently been proposed [4] to efficiently handle the complementarity conditions (8) and (9). Unlike primal-dual IP methods, the NC approach does not require that the strict positivity conditions  $(s, z, \pi, \hat{v}) > 0$  be verified at every single point of the iterative process. It has been derived from recently developed techniques for solving linear and nonlinear *complementarity problems* [5, 6, 7, 8, 9].

The most attractive feature of the NC approach is that it reformulates the optimality conditions (4) to (9) as a nonlinear system of equations, thus allowing for a Newton-type method to be used. The non-negativity conditions  $(s, z, \pi, \hat{v}) \geq 0$  are automatically satisfied at the limit point, without imposing additional conditions during the iterative process. The equation-reformulation is obtained by handling each complementarity condition through a function  $\psi : \mathbb{R}^2 \mapsto \mathbb{R}$  that holds the property

$$\psi(a, b) = 0 \iff ab = 0, \quad a \geq 0, \quad b \geq 0. \quad (17)$$

Notice, for instance, that the complementarity condition

$$s_i \pi_i = 0, \quad s_i \geq 0, \quad \pi_i \geq 0, \quad (18)$$

seen in (8), is equivalent to the nonlinear equation

$$\psi(s_i, \pi_i) = 0. \quad (19)$$

Any function that holds the property (17), such as

$$\psi(a, b) = \sqrt{a^2 + b^2} - (a + b), \quad (20)$$

$$\psi(a, b) = \frac{1}{2} \min\{a, b\}, \quad (21)$$

$$\psi(a, b) = \frac{1}{2} ((ab)^2 + \min\{0, a\} + \min\{0, b\}), \quad (22)$$

is said to be a NC-function. The function in (20) was first introduced by Fischer [5] in 1992, and has attracted a lot

of attention from researchers. Notice, on one hand, that this function is differentiable everywhere but in the origin. On the other hand, its simple smooth approximation

$$\psi_\mu(a, b) = \sqrt{a^2 + b^2 + 2\mu} - (a + b), \quad (23)$$

proposed by Kanzow [7], is differentiable everywhere for any  $\mu > 0$ . The zeros of (23) are characterized by

$$\psi_\mu(a, b) = 0, \iff ab = \mu, \quad a > \mu, \quad b > \mu, \quad (24)$$

and when we force  $\mu$  to go to zero the property (24) closely approximates the property (17).

In this paper, we introduce a Jacobian smoothing nonlinear complementarity algorithm for solving (1), derived from smooth and non-smooth equation-reformulations of the KKT conditions. Section 2 presents the equation-reformulations and introduces the Jacobian smoothing NC algorithm. Section 3 details the implemented algorithms, and Section 4 shows the numerical results obtained as the algorithms are applied to the OPF problems described in Section 1. Some conclusions in Section 5 close the paper.

## 2 THE OPF NONLINEAR EQUATIONS REFORMULATION

By exploiting the property (17), it is readily seen the equivalence between the complementarity conditions (8) and (9) and nonlinear equations systems, in the form

$$S\pi = 0, \quad s \geq 0, \quad \pi \geq 0, \iff \psi(s, \pi) = 0, \quad (25)$$

$$Z\hat{v} = 0, \quad z \geq 0, \quad \hat{v} \geq 0, \iff \psi(z, \hat{v}) = 0. \quad (26)$$

It is then obvious that the KKT conditions (4) to (9) can be reformulated as the nonlinear equations system

$$F(y) = \begin{pmatrix} -g(x) \\ s + z - \bar{h} + \underline{h} \\ h(x) + z - \bar{h} \\ \nabla f(x) - J_g(x)^T \lambda + J_h(x)^T v \\ \psi(s, \pi) \\ \psi(z, \hat{v}) \end{pmatrix} = 0 \quad (27)$$

where  $y = (x, s, z, \pi, v, \lambda)^T$  is the vector of all variables. The advantage of the reformulation (27) is that, unlike the KKT conditions (4) to (9), it can be solved iteratively by well-known methods [10]. The conditions  $(s, z, \pi, \hat{v}) \geq 0$  are automatically assured by the NC-function at the limit point of the iterative process.

As it is well known that with a poor initial estimate Newton's method may diverge, we solve  $F(y) = 0$  by a Newton-type method with step size control to improve the convergence. The basic step of the solution algorithm is: Given a solution estimate  $y_k$ , solve the linear system

$$J_F(y_k) \Delta y = -F(y_k) \quad (28)$$

for the correction term  $\Delta y$ , and compute a new solution estimate from

$$y_{k+1} = y_k + \alpha_k \Delta y. \quad (29)$$

The matrix  $J_F(y_k)$  is the Jacobian of  $F(y)$  evaluated at  $y_k$ , and the scalar  $\alpha_k \in (0, 1]$  is the step size parameter that enhances the convergence.

### 2.1 Nonsmooth Nonlinear Complementarity Algorithm

If the NC-function used in (27) is non-smooth, like the Fischer-Burmeister function in (20), then  $F(y)$  also is non-smooth and we cannot apply the classical Newton method in order to solve  $F(y) = 0$ . On the other hand, we can apply a non-smooth Newton method based, for example, on Clarke's [11] generalized Jacobian  $\partial F(y)$  of  $F(y)$ . In such a case, the non-smooth Newton method solves at each iteration the generalized Newton equation

$$V_k d_k = -F(y_k) \quad (30)$$

for the search direction  $d_k$ , where  $V_k \in \partial F(y_k)$ . A new solution estimate is computed from

$$y_{k+1} = y_k + \alpha_k d_k. \quad (31)$$

We refer the interested reader to [12] for further details on non-smooth Newton methods.

### 2.2 Smooth Nonlinear Complementarity Algorithm

An alternate way to deal with the non-smoothness of  $\psi(a, b)$ , and hence of  $F(y)$ , is to replace this function by a smooth approximation, as the one proposed by Kanzow in [7]—the function  $\psi_\mu(a, b)$  in (23). The paper [9] by Burke and Xu initiated much of the recent research in the area of smoothing methods for solving complementarity problems. A list of related works is presented in [13].

If in order to reformulate the KKT conditions as a nonlinear equations system we use the smooth function in (23) instead of the non-smooth one in (20), then we obtain the smooth equations system

$$F_\mu(y) = \begin{pmatrix} -g(x) \\ s + z - \bar{h} + \frac{h}{\mu} \\ h(x) + z - \bar{h} \\ \nabla f(x) - J_g(x)^T \lambda + J_h(x)^T v \\ \psi_\mu(s, \pi) \\ \psi_\mu(z, \hat{v}) \end{pmatrix} = 0. \quad (32)$$

By forcing the smoothing parameter  $\mu$  to go to zero the equations system (32) closely approximates the equations system (27). The advantage of the smooth reformulation (32), though it is an approximation, as compared with (27), is that we can apply the standard Newton method to solve it. In such a case, we have to solve at each iteration the smoothing Newton equation

$$J_{F_\mu}(y_k) \Delta y = -F_\mu(y_k) \quad (33)$$

for the correction term  $\Delta y$  and compute a new solution estimate from (29). The matrix  $J_{F_\mu}(y_k)$  is the Jacobian of  $F_\mu(y)$  evaluated at  $y_k$  and, unlike  $J_F(y_k)$ , is smooth.

### 2.3 Jacobian Smoothing Nonlinear Complementarity Algorithm

The Jacobian smoothing NC algorithm described here is derived from ideas of Kanzow and Pieper presented in [13], and is the main focus of this paper. To obtain

the search directions, the Jacobian smoothing method try to solve in each iteration the mixed Newton equation

$$J_{F_\mu}(y_k) d_k = -F(y_k). \quad (34)$$

Notice that the linear system (34) is a mixture between the non-smooth Newton equation (28) and the smoothing Newton equation (33). It uses the unperturbed right-hand side from (28),  $-F(y_k)$ , but the smooth matrix from (33),  $J_{F_\mu}(y_k)$ . This is one of the major differences between this algorithm and the one presented in [4, 14].

Kanzow and Pieper comment in [13] that despite the fact the Jacobian smoothing methods are often viewed as a variation of smoothing methods, they seem to be much closer to non-smooth Newton methods than to smoothing methods since they do not try to follow any smoothing path. Instead, they also try to solve the unperturbed problem (27) directly by replacing the matrix  $V_k \in \partial F(y_k)$  in (30) by a suitable approximation  $J_{F_\mu}(y_k)$ .

## 3 THE IMPLEMENTED OPF ALGORITHM

The main steps of the Jacobian smoothing algorithm are as follows:

*Step 0:* Set  $k = 0$ , choose  $\mu_0 > 0$  and a starting point  $y_0 = (s_0, z_0, \pi_0, v_0, x_0, \lambda_0)$ .

*Step 1:* Form the mixed Newton equation at the point  $y_k$ ,

$$J_{F_\mu}(y_k) d_k = -F(y_k),$$

and solve for the search direction  $d_k$ .

*Step 2:* Perform a line search along  $d_k$  to obtain the step  $\alpha_k$ , and compute a new solution estimate from

$$y_{k+1} = y_k + \alpha_k d_k.$$

*Step 3:* If  $y_{k+1}$  satisfies the convergence criteria, then STOP. If not, then set  $k \leftarrow k + 1$ , compute the parameter  $\mu_k < \mu_{k-1}$ , and return to *Step 1*.

If the search direction  $d_k$  that is computed in Step 1 is not a “good” descent direction, according to the test

$$F(y_k)^T J_{F_\mu}(y_k) d_k \leq -\rho \|d_k\|^p, \quad \rho > 0, \quad p > 2, \quad (35)$$

then we resort to the *steepest descent direction*, that is, we set  $d_k = -J_{F_\mu}(y_k)^T F(y_k)$  to ensure that the search direction is “sufficiently” downhill.

### 3.1 Computing the Step Sizes

The step size  $\alpha_k$  is obtained through a line search along the search direction  $d_k$ . Such a line search is based on a merit function  $\phi(y)$  that measures the progress of the sequence of iterates  $\{y_k\}$  toward a solution to (27). In this paper, we consider the simple natural merit function

$$\phi(y) = \frac{1}{2} \|F(y)\|_2^2. \quad (36)$$

It is clear that the global minimum to  $\phi(y)$  (which has zero optimal value) is a solution to  $F(y) = 0$ . Hence, given a solution estimate  $y_k$  and a search direction  $d_k$ , the step  $\alpha_k$

along  $d_k$  can be obtained as an approximate solution to the unconstrained minimization problem in one variable

$$\min_{\alpha \in (0,1]} \phi(y_k + \alpha d_k). \quad (37)$$

Nowadays, however, it is customary to perform “inexact” line searches, as the Armijo-type rule [10]:

- Given the scalars  $\gamma \in (0, 1]$  and  $\beta \in (0, 1)$ , find the smallest integer  $m \in \{0, 1, 2, \dots\}$  that satisfies the test

$$\phi(y_k + \beta^m d_k) \leq (1 - \gamma \beta^m) \phi(y_k). \quad (38)$$

- Then, let  $\alpha_k = \beta^m$ .

The “inexactness” of this procedure is typified by a small value of  $\gamma$  (typically,  $\gamma = 10^{-4}$ ) and a large value of  $\beta$  (typically,  $\beta = 0.99$ ).

### 3.2 Reducing the Smoothing Parameter

In order to reduce the smoothing parameter  $\mu_k$ , first we compute the measure

$$\rho_k = |s_k^T| |\pi_k| + |z_k^T| |\hat{v}_k|, \quad (39)$$

which “mimics” the residual of the complementarity gap. If the iterates converge to an optimum, then  $\{\rho_k\} \rightarrow 0$ . Thus, we propose to compute  $\mu_{k+1}$  based on an expected cut of the average complementarity gap, as follows:

$$\mu_{k+1} = \min \left\{ 0.2 \frac{\rho_k}{2p}, 0.9 \mu_k \right\}. \quad (40)$$

### 3.3 Testing the Convergence

The iterative process is considered converged if

$$\begin{aligned} & \|g(x_k)\| \leq \epsilon_1 \\ & \max \{ \max \{ \underline{h} - h(x_k) \}, \max \{ h(x_k) - \bar{h} \} \} \leq \epsilon_1 \\ & \frac{\|\nabla f(x_k) - J_g(x_k)^T \lambda_k + J_h(x_k)^T v_k\|_\infty}{1 + \|x_k\|_2 + \|\lambda_k\|_2 + \|v_k\|_2} \leq \epsilon_1 \\ & \frac{\max \{ |\min \{ s_k, \pi_k \}|, |\min \{ z_k, \hat{v}_k \}| \}}{1 + \|x_k\|_2} \leq \epsilon_2 \end{aligned}$$

Otherwise, either the iterative process is stuck at some point other than a minimum or it has diverged. Typical convergence tolerances are  $\epsilon_1 = 10^{-4}$  and  $\epsilon_2 = 10^{-6}$ .

## 4 SOME NUMERICAL RESULTS

In order to assess the computational performance of the smooth (see Section 2.2) and Jacobian smoothing (see Section 2.3) NC algorithms, we apply them to solve three different OPF problems: (a) minimum active power loss, (b) maximum system loadability, and (c) minimum load shedding. The constraints include power balance equations, limits on voltage magnitudes, on transformers tap ratios, on generators reactive power outputs, and on shunt var sources. The results obtained with the NC algorithms are compared with those obtained with the standard primal-dual and the predictor-corrector IP algorithms [15].

Implementations of the algorithms were developed both in Matlab and in Fortran 77, but the current version of the Fortran 77 codes only solve the loss minimization problem. In order to solve the indefinite linear systems in Step 1 of the optimization algorithms, we have utilized the public domain linear system solver UMFPACK of Timothy Davis and Iain Duff [16]. All test runs have used the same initialization heuristic (the one suggested in [15]) and set of parameters:  $\mu_0 = 0.01$ ,  $\gamma = 0.0001$ ,  $\beta = 0.75$  and  $\epsilon_1 = 10^{-4}$ . The results were produced on a Pentium III 500 MHz with 128 MBytes of RAM, running under Linux. All solution times reported are in seconds. They include the whole processing time except data input and solution output.

The tests are carried out on seven power networks that range in size from 30 to 2098 buses, the largest system in this set being a modified reduced system derived from the Brazilian South/Southeast power network. Table 1 displays, for each power system, the total number of buses,  $\mathcal{N}$ , the number of *generator* buses,  $\mathcal{G}$ , the number of load buses *eligible* for shunt var control,  $\mathcal{E}$ , the total number of *branches* (transmission lines and transformers),  $\mathcal{B}$ , and the number of *transformers* with LTC device,  $\mathcal{T}$ . Also given are the number of primal variables,  $n$ , the number of equality constraints,  $m$ , and the number of inequality constraints with upper and lower bounds,  $p$ . Table 2 displays the total system active and reactive power loads, and the transmission losses for the base condition, that is, prior to the application of any optimization procedure.

**Table 1:** Sizes of the power systems and of the OPF problem (1)

System	$\mathcal{N}$	$\mathcal{G}$	$\mathcal{E}$	$\mathcal{B}$	$\mathcal{T}$	$n$	$m$	$p$
1 IEEE	30	6	5	41	4	63	48	45
2 IEEE	57	7	5	80	10	123	101	79
3 IEEE	118	54	12	186	9	244	169	193
4 IEEE	300	69	12	411	50	649	518	431
5 MEX	256	68	23	376	50	561	430	387
6 TEST	555	126	46	787	85	1194	937	812
7 BRAZ	2098	169	426	3283	239	4434	3600	2932

**Table 2:** Initial loading (active and reactive) and transmission losses of the power systems.

System	$P$ (MW)	$Q$ (MVA <sub>r</sub> )	$P_L$ (MW)
1 IEEE	283.40	126.20	17.62
2 IEEE	1250.80	336.40	27.99
3 IEEE	3668.00	1438.00	129.88
4 IEEE	23246.87	7787.97	408.43
5 MEX	10395.80	3382.00	210.17
6 TEST	33642.67	11169.98	617.78
7 BRAZ	19570.60	10746.51	1173.60

### 4.1 Active Loss Minimization

Tables 3 and 4 display the results obtained with the Fortran 77 codes to solve the active loss minimization problem. They show the number of iterations (iter), the total number of merit function evaluations ( $\phi$ -eval), the number of full/damped Newton steps taken (f / d), and the CPU times (time) in seconds. Also given are the number of iterations and CPU times for the standard primal-

dual IP method (Table 3) and for the successful predictor-corrector IP method (Table 4). The NC algorithms have used the same initialization used by the IP algorithms. We can see from these tables that, as far as the CPU times are concerned, the best performance overall has been that of the Jacobian smoothing NC algorithm (Section 2.3), which was able to optimize all test systems in 28.29 seconds. However, its performance has been quite close to that of the smoothing NC algorithm (Section 2.2), in terms of both number of iterations ( $94 \times 92$ ) and CPU times ( $28.29 \times 29.30$ ).

**Table 3:** The smooth NC algorithm (Section 2.2) and the standard primal-dual IP algorithm applied to loss minimization

Test System	S-NC				PD-IP	
	iters	$\phi$ -eval	f / d	time	iters	time
1 IEEE	8	6	5/3	0.46	10	0.58
2 IEEE	10	25	7/3	0.70	13	0.87
3 IEEE	12	19	7/5	1.16	15	1.34
4 IEEE	13	21	10/3	2.33	17	2.78
5 MEXI	19	72	8/11	2.86	16	2.23
6 TEST	16	16	11/5	4.90	19	5.46
7 BRAS	14	48	8/6	16.89	$\alpha_5 \approx 0$	
Total	92	207	56/36	29.30		

The entry  $\alpha_5 \approx 0$  in the last column of Table 3 shows that the standard primal-dual IP algorithm (PD-IP) has failed to optimize the power system with 2098 buses. The reason for this failure is that poorly centered iterates (there are indexes  $i$  and  $j$  such that  $s_i \pi_i \ll s_j \pi_j$ ) cause the step length  $\alpha_k$  to be very close to zero at iteration 5. Then, the primal-dual IP algorithm cannot progress any further toward a solution point. Poorly centered iterates usually happen in problems with strongly active limits, as occur in heavily loaded systems and systems with bottlenecks in its transmission network.

The predictor-corrector IP algorithm (PC-IP) was very successful in optimizing all systems, as shown in the last two columns of Table 4. Although the predictor-corrector IP algorithm required less iterations to optimize the whole set of systems, both NC algorithms were more efficient to solve the largest problem. Surprisingly, the Jacobian smoothing NC algorithm solved the 2098 bus system in 12 iterations and 14.47 seconds, in contrast to 23 iterations and 25.54 seconds of the predictor-corrector IP algorithm. However, this superior performance of the NC algorithms has not been observed in other test runs, and this special case should not be taken as conclusive.

**Table 4:** The Jacobian smoothing NC algorithm (Section 2.3) and the predictor-corrector IP algorithm applied to loss minimization

Test System	JS-NC				PC-IP	
	iters	$\phi$ -eval	f / d	time	iters	time
1 IEEE	7	3	6/1	0.39	7	0.37
2 IEEE	9	9	7/2	0.59	8	0.57
3 IEEE	11	23	8/3	1.07	8	0.79
4 IEEE	19	58	12/7	3.41	10	1.72
5 MEXI	18	36	14/4	2.71	10	1.60
6 TEST	18	51	11/7	5.65	12	3.70
7 BRAS	12	52	6/6	14.47	23	25.54
Total	94	232	64/30	28.29	78	34.29

It is important to mention that all algorithms have converged to the same solution. The computational effort per iteration in the IP and NC algorithms is essentially the same, as long as the NC algorithms do not perform too many merit functions evaluations. This is, in turn, related to the “quality” of the search directions  $d_k$ . It should be pointed out here that the conditional test (35), for assessing the “quality” of the search directions, was not used to obtain the results presented in this paper. Then, it is reasonable to expect to cut down the CPU times of the NC codes by implementing more elaborate merit functions and schemes for ensuring “good” descent directions, such as a Levenberg-Marquardt type scheme [10].

Table 4 shows that the NC algorithm has performed a total of 232 merit function evaluations for a total of 30 damped iterations out of 94. This gives an average of 7 to 8 merit function evaluations per damped iteration. However, such an average is not very meaningful, because we have observed from extensive simulations that some damped iterations performs very little merit function evaluations, while others perform the allowed maximum number of trial steps per iteration ( $m$  in the conditional test (38)), which is set to 15. In other words, the NC algorithms, in the current version of the code, has used non-descent search directions in some iterations.

#### 4.2 Maximum Loadability

Tables 5 and 6 show the results obtained with the Matlab code for computing the maximum loadability of a power system. The results shown below consider load increases at the PQ buses only. However, the code also allows the specification of any subset of buses or even a single bus for load increase. Recall from Section 1 that only constant power loads are modeled. Table 5 shows the loading parameter  $\sigma$ , the total system power loads at the critical point, and the corresponding active losses. The largest load increase, of 74.10%, has been observed with the IEEE 118-buses system, and the smallest one, of 2.34%, with the IEEE 300-buses system. This small increase in the loading of the IEEE 300 buses-system is no surprise since we set, on purpose, very tight operating limits for this system, and specified a small number of load buses for shunt reactive power compensation. As expected, power losses are very high at the critical point.

**Table 5:** Maximum loading (active and reactive) and transmission losses at the critical point

System	$\sigma$	$P$ (MW)	$Q$ (MVar)	$P_L$ (MW)
1 IEEE	0.5916	364.75	164.36	31.80
2 IEEE	0.4412	1439.98	396.58	53.33
3 IEEE	0.7410	4729.88	1827.03	282.17
4 IEEE	0.0234	23621.88	7912.29	459.88

Table 6 shows the number of iterations for the Jacobian smoothing NC algorithm (JS-NC), and for the standard primal-dual (PD-IP) and the predictor-corrector (PC-IP) interior-point algorithms. On one hand, as far as the number of iterations are concerned, the best performance has

been that of the predictor-corrector IP algorithm. On the other hand, the standard primal-dual IP method failed to optimize all the systems; poorly centered iterations make the step sizes close to zero around the 4th iteration. The Jacobian smoothing NC algorithm was able to optimize all the systems. Table 6 also shows the number of active limits on voltage magnitudes ( $V$ ), tap ratios ( $t$ ), reactive power outputs of generators ( $Q$ ), and shunt reactive power compensations ( $b^{sh}$ ). Notice the high number of active limits at the critical point, as expected. For instance, the IEEE 300-buses system has 102 active voltage limits out of 300, and 45 reactive power limits out of 69.

**Table 6:** Number of iterations and active limits: Maximum Loadability Problem

System	JS-NC	PD-IP	PC-IP	$V$	$t$	$Q$	$b^{sh}$
1 IEEE	12	Failed	10	4	1	6	4
2 IEEE	15	Failed	12	17	2	5	5
3 IEEE	18	Failed	14	43	7	32	11
4 IEEE	22	Failed	17	102	6	45	4

Table 7 shows the maximum loading of the systems as the voltage limits at the PQ buses are increased by 5%, and generators reactive power output limits by 10%, since they had been set very tight. By comparing the  $\sigma$ -columns of Tables 7 and 6, we can see the increase in the loading of the systems. For instance, the loadability of the IEEE 300-buses system has increased from 2.34% to 3.06%. This quite small loadability should not be interpreted as the system being close to a collapse point, but being close to violate the specified operating limits. As expected, the higher is the loading the higher are the power losses. The Table 8 shows the number of iterations and active limits at the critical point. As the system loading has increased, the number of active limits has increased as well. For instance, the number of active voltage limits has increased from 102 to 143, that is, to almost half the buses.

**Table 7:** Maximum loading (active and reactive) and transmission losses at the critical point, with extended limits

System	$\sigma$	$P$ (MW)	$Q$ (MVar)	$P_L$ (MW)
1 IEEE	0.6753	376.25	169.75	33.74
2 IEEE	0.5493	1486.33	411.32	63.57
3 IEEE	0.8525	4889.69	1885.58	323.45
4 IEEE	0.0306	23738.03	7950.80	493.04

**Table 8:** Number of iterations and active limits: Maximum Loadability Problem with extended limits

System	JS-NC	PD-IP	PC-IP	$V$	$t$	$Q$	$b^{sh}$
1 IEEE	13	Failed	12	14	1	5	4
2 IEEE	15	Failed	11	18	10	7	5
3 IEEE	20	Failed	15	47	3	32	11
4 IEEE	14	Failed	15	143	10	36	5

### 4.3 Minimum Load Shedding

Table 9 shows the results obtained with the Matlab code to solve the minimum load shedding problem. In order to obtain the test problems, rather than simulating contingencies that lead the systems to collapse, we have increased their loads in all buses by 90%. This increase of 90% has been chosen based on the maximum loadings

indicated in Tables 5 and 7. Notice that the less loaded system, IEEE 118-buses system, allows a load increase of only 74.10%. It should be pointed out here that the load increase of 90% has been applied to all system buses, while the load shedding is allowed only at the PQ buses. This explains why the algorithms were unable to solve the IEEE 300-buses system. Its solution space is empty, that is, it is impossible to obtain a feasible operating point for this system if we increase by 90% the loads of all buses and allow load shedding only at the PQ buses. Table 10 shows the number of iterations and active limits.

**Table 9:** Minimum load shedding (active and reactive) and transmission losses at the critical point.

System	$\sigma$	$P$ (MW)	$Q$ (MVar)	$P_L$ (MW)
1 IEEE	0.6764	361.74	156.88	31.36
2 IEEE	1.1685	1424.46	336.31	53.17
3 IEEE	0.8157	4748.25	1918.52	300.29
4 IEEE	-	-	-	-

**Table 10:** Number of iterations and active limits: Minimum Load Shedding Problem

System	JS-NC	PC-IP	$V$	$t$	$Q$	$b^{sh}$
1 IEEE	14	10	6	0	5	4
2 IEEE	11	15	16	4	5	3
3 IEEE	15	16	42	3	37	11
4 IEEE	Failed	Failed	-	-	-	-

Tables 11 and 12 show the results obtained when the voltage limits at the PQ buses are increased by 5%, and the reactive power of the generators are increased by 10%. As expected, the amount of load shedded has decreased with the increase in some of the constraints limits. With the above mentioned increase in voltage and reactive power limits, and a load increase of 20% only in the IEEE 300-buses system, it was possible to obtain a feasible operating point for this system.

**Table 11:** Minimum load shedding (active and reactive) and transmission losses at the critical point.

System	$\sigma$	$P$ (MW)	$Q$ (MVar)	$P_L$ (MW)
1 IEEE	0.6062	380.09	165.49	34.57
2 IEEE	1.0853	1492.28	357.89	65.42
3 IEEE	0.7623	4893.82	1971.85	327.67
4 IEEE	0.2151	23756.52	7973.21	493.56

**Table 12:** Number of iterations and active limits: Minimum Load Shedding Problem

System	JS-NC	PC-IP	$V$	$t$	$Q$	$b^{sh}$
1 IEEE	11	11	10	0	5	4
2 IEEE	18	13	27	6	4	3
3 IEEE	22	15	38	4	35	12
4 IEEE	25	16	147	9	35	7

## 5 CONCLUSIONS

The paper has described in great detail a Jacobian smoothing nonlinear complementarity method for solving nonlinear optimal power flows. The distinctive feature of the NC approach is that it reformulates optimization problems as nonlinear equations systems, which can then be solved by well-known methods. The Jacobian smoothing

NC algorithm has been applied with considerable success to three OPF problems: active loss minimization, maximum system loadability and minimum load shedding. Its performance, in terms of robustness and CPU times, has been comparable with that of the successful and widely used predictor-corrector IP technique. We believe that further research is worthwhile along the lines presented in this work, both from the theoretical and experimental point of view. In particular, the implementation of procedures that ensure good descent search directions, in order to reduce the number of merit functions evaluations and, more importantly, ensure convergence of the algorithm.

### ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support from CNPq in Brazil and NSERC in Canada.

### REFERENCES

- [1] S. Granville, J. C. O. Mello, and A. C. G. Melo, "Application of interior point methods to power flow unsolvability," *IEEE Trans. on Power Systems*, vol. 11, no. 2, pp. 1096–1103, May 1996.
- [2] G. D. Irisarri, X. Wang, J. Tong, and S. Mokhtari, "Maximum loadability of power systems using interior point non-linear optimization method," *IEEE Trans. on Power Systems*, vol. 12, no. 1, pp. 162–172, Feb. 1997.
- [3] C. A. Cañizares, "Applications of optimization to voltage collapse analysis," 1998, Panel Session: "Optimization Techniques in Voltage Collapse Analysis," IEEE/PES Summer Meeting, San Diego.
- [4] G. L. Torres and V. H. Quintana, "Optimal power flow by a nonlinear complementarity method," in *Proc. of IEEE 21st Power Industry Computation Conference*, Santa Clara, CA, USA, May 1999, pp. 211–216. Also in *IEEE Trans. on Power Systems*, vol. 15, no. 3, pp. 1028–1033, Aug. 2000.
- [5] A. Fischer, "A special Newton-type optimization method," *Optimization*, vol. 24, pp. 269–284, 1992.
- [6] C. Kanzow, "Some equation-based methods for the nonlinear complementarity problem," *Optimization Methods and Software*, vol. 3, pp. 327–340, 1994.
- [7] C. Kanzow, "Some noninterior continuation methods for linear complementarity problems," *SIAM J. on Matrix Analysis and Applications*, vol. 17, no. 4, pp. 851–868, Oct. 1996.
- [8] T. De Luca, F. Facchinei, and C. Kanzow, "A semismooth equation approach to the solution of nonlinear complementarity problems," *Mathematical Programming*, vol. 75, pp. 407–439, 1996.
- [9] J. V. Burke and S. Xu, "The global linear convergence of a non-interior path-following algorithm for linear complementarity problems," *Mathematics of Operations Research*, vol. 23, pp. 719–734, 1999.
- [10] J. E. Dennis Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM Classics In Applied Mathematics, Philadelphia, 1996.
- [11] F. H. Clarke, *Optimization and Nonsmooth Analysis*, SIAM, 1990.
- [12] L. Qi and D. Sun, "A nonsmooth version of Newton's method," *Mathematical Programming*, 1993.
- [13] C. Kanzow and H. Pieper, "Jacobian smoothing methods for general nonlinear complementarity problems," Technical Report, Institute of Applied Mathematics, University of Hamburg, Hamburg, Germany, Oct. 1997.
- [14] V. H. Quintana and G. L. Torres, "Optimal power flow by interior and non-interior point methods for nonlinear programming," in *Proc. of the VI SE-POPE*, Salvador, Bahia, Brasil, May 1998, Paper IP-63 CDROM.
- [15] G. L. Torres and V. H. Quintana, "An interior point method for nonlinear optimal power flow using voltage rectangular coordinates," *IEEE Trans. on Power Systems*, vol. 13, no. 4, pp. 1211–1218, Nov. 1998.
- [16] T. A. Davis and I. S. Duff, "An unsymmetric-pattern multifrontal method for sparse LU factorization," *SIAM J. on Matrix Analysis and Applications*, vol. 19, no. 1, pp. 140–158, 1997.