An Interior Point Algorithm for Nonlinear Programming*

Neculai Andrei

Research Institute for Informatics

8-10, Averescu Avenue, Bucharest

ROMANIA

e-mail: nandrei@u3.ici.ro

Abstract:

This paper describes an interior point algorithm for solving nonlinear programming problems. The approach we consider here is a classical one. It consists of transforming the original problem into one with only equality constraints, the inequalities being placed into the objective through a logarithmic barrier function. The KKT optimality conditions are solved by the Newton method. The direction of moving is given by a linear algebraic system, which must be solved at each iteration. The step length computation, which is the critical point of the algorithm, is based on a merit function which determines the values of the barrier parameter. This ensures the descendance character of the search direction. The conditions for step length computation include: the boundary of variables, the positivity of the slack and dual variables, the centrality of iterations, the correlation between the speed of decreasing the pure optimality conditions and the transversality conditions, as well as the Wolfe conditions.

A crude implementation of the algorithm shows the performance of this approach on a number of nonlinear problems from the Hock-Schittkowski set of problems. Comparisons with a modified penalty-barrier algorithm implemented as a SPENBAR package show that the method is efficient on at least some classes of nonlinear models.

Dr. Neculai Andrei graduated in Electrical Engineering from the Polytechnical Institute of Bucharest and took his Ph.D degree for his contribution to digraph approach of large-scale sparse linear dynamic systems in 1973 and 1984 respectively. He is author of a number of published papers and technical reports in the area of mathematical modelling and optimization. He is author of the books: *Sparse Matrices and their Applications*, Technical Publishing House, Bucharest,1983, and *Sparse Systems-Digraph Approach of Large- Scale Linear Systems Theory*, TUV Verlag-Rheinland, Köln, Germany, 1985. His main current scientific interests centre on languages for mathematical programming problems modelling, large-scale linear and non-linear optimization, interior point methods, penalty and barrier approaches, etc. Dr. Andrei is a founding member of the Operations Research Society of Romania, member of the Editorial Board of Computational Optimization and Applications -an International Journal, and of IFAC/WG2 Control Applications of Optimizations.

1. Introduction

Interior point methods for nonlinear programming are among the most active research areas in mathematical programming. This is motivated by the impressive high quality theoretical results and computational performance of these methods for linear and quadratic programming, as well as for linear constrained optimization. We can mention here, for example, the papers of [Andersen and Andersen, 1997], [Andrei, 1998b], [Carpenter, Lustig, Mulvey and Shanno, 1993], [Gondzio, 1996], [Lustig, Marsten and Shanno, 1990, 1991, 1992, 1994], [Vanderbei,1990, 1997], [Ye, 1997].

Having in view these remarkable results, it is natural that the researches using interior point ideas are directed to the nonlinear programming area [Kortanek, Potra and Ye, 1991], [Goldfarb, Liu and Wang, 1991], [Wright, 1992, 1998].

*Special thanks are due to the Romanian Academy for supporting this research work through Grant No. GAR-11 / 16.07.1998.

We are interested in solving the following nonlinear programming problem:

$$\begin{aligned} & min f(x) \\ & subject\ to: \\ & g(x) = 0, \\ & h(x) \geq 0, \\ & l \leq x \leq u, \end{aligned} \tag{1.1}$$

where $f : R^n \to R$, $g : R^{me} \to R$ and $h : R^m \to R$ are supposed to be twice continuous differentiable. The bounds l and u may have any values with $l < u$, in practice these could be $\pm\infty$. For solving this general nonlinear programming problem we have a number of methods and algorithms which could be classified as: Direct Methods, Methods based on KKT Optimality Conditions, Penalty Methods, and Interior Point Methods. The interior point methods have recently been the subject of a number of papers. The formulation and theory of the Newton interior point method for nonlinear programming are given in [El-Bakry, Tapia, Tsuchiya and Zhang, 1996]. Gay, Overton and Wright [1997] consider a primal-dual method for nonconvex programming in which the inequality constraints are introduced in the objective function by means of the classical logarithmic barrier function, and the equality constraints are considered in a sequential quadratic programming mechanism. Vanderbei and Shanno [1997] transform the original inequality problem into an equality one by adding slack to each of the inequality constraints, for which the first order conditions for a minimum are solved by means of a Newton strategy. A special method for choosing the step length along the Newton direction is introduced. This is based on a merit function. An extension of the LOQO package to considering the nonlinear function, interfaced with AMPL [Fourer, Gay and Kernighan, 1993] and GAMS [Brooke, Kendrick and Meeraus, 1992], is illustrated on the Hock and Schittkowski suite, Mittelmann's quadratic programming set, as well as on a number of 8 large-scale real-world problems. Some comparisons and performances of LOQO with MINOS [Murtagh and Saunders, 1995] and LANCELOT [Conn, Gould and Toint, 1992] are presented. Byrd, Hribar and Nocedal [1997] incorporate into the interior point method two powerful tools for solving nonlinear problems: the sequential quadratic programming and the trust region techniques. The NITRO package is presented and its performance on some numerical examples is illustrated. Some other primal-dual interior methods for nonconvex nonlinear optimization problems are discussed in [Argaez and Tapia, 1997], [Byrd, Gilbert and Nocedal, 1996], [Conn, Gould and Toint, 1996], [Forsgren and Gill, 1996], etc.

The algorithms are based on the classical approach: a minimizing direction is computed and then a step length on this direction is being considered. When the problem is a *convex program* the step length can be determined by standard line search methods, which are direct extensions of interior point methods for linear programming. In this case the step generated by the solution of the primal-dual equations is a descent direction for different merit functions, thus ensuring the convergence of the corresponding algorithms. For *non-convex programming*, the method of computing the step length at each iteration is more complex. In fact, the notable differences between algorithms are in the manner in which the step length is computed. Thus, for example, Vanderbei and Shanno [1997] introduce into the merit function a parameter which is updated as soon as the computed direction is not a descent direction. On the other hand, Gay, Overton and Wright [1997] describe a heuristic procedure combined with a simple Armijo-like rule to determine the step length.

In this paper we shall present an interior point predictor corrector algorithm for solving (1.1) which is based on the theoretical developments given in [El-Bakry, Tapia, Tsuchiya and Zhang, 1996]. Basically, for the considered problem the KKT optimality conditions are specified. Using the Newton method we meet the optimality conditions with respect to the moving direction. Considering a suitable merit function the updating conditions for the barrier parameter are established. The mechanism for step length determination is based on the Wolfe conditions combined with the interior point centrality condition and the limitation of the convergence to zero of the transversality conditions against the "pure primal-dual" optimality condition. The paper is organized as follows. Section 2 presents the KKT optimality conditions. The Newton system and the Newton direction are considered in Sections 3 and 4, respectively. The merit function is introduced in Section 5. Section 6 is devoted to computing the step length on the Newton direction. The next Sections present the primal-dual algorithm and some numerical examples.

2. KKT Optimality Conditions

Considering the slack variables $s \in R^m$, associated with the functional inequality constraints and $w, v \in R^n$ associated with the simple bounds of variable x, the problem (1.1) can be reformulated as:

$$\begin{gathered}\min f(x)\\ subject\ to:\\ g(x)=0,\\ h(x)-s=0,\\ x-w=l,\\ x+v=u,\\ s\geq 0, w\geq 0, v\geq 0.\end{gathered} \tag{2.1}$$

Now, we eliminate the inequality constraints in (2.1) by placing them into a barrier term, obtaining the problem:

$$\begin{gathered}\min f(x)-\mu\sum_{i=1}^{m}\log s_i-\mu\sum_{j=1}^{n}\log w_j-\mu\sum_{j=1}^{n}\log v_j\\ subject\ to:\\ g(x)=0,\\ h(x)-s=0,\\ x-w-l=0,\\ x+v-u=0,\end{gathered} \tag{2.2}$$

where the objective function in (2.2) is the classical Fiacco-McCormick [1968] logarithmic barrier function.

The Lagrangian for this problem is:

$$\begin{gathered}L(x,s,w,v,y,z,p,q,\mu)=\\ f(x)-\mu\sum_{i=1}^{m}\log s_i-\mu\sum_{j=1}^{n}\log w_i-\mu\sum_{j=1}^{n}\log v_i-\\ y^Tg(x)-z^T(h(x)-s)-p^T(x-w-l)+q^T(x+v-u),\end{gathered} \tag{2.3}$$

where y, z, p and q are the dual variables of the corresponding dimensions. With these, the first order optimality conditions are:

$$\begin{gathered}\nabla f(x)-\nabla g(x)^Ty-\nabla h(x)^Tz-p+q=0,\\ g(x)=0,\\ h(x)-s=0,\\ x-w-l=0,\\ x+v-u=0,\\ SZe-\mu e=0,\\ WPe-\mu e=0,\\ VQe-\mu e=0,\end{gathered} \tag{2.4}$$

where the matrices S, Z, W, P, V and Q are diagonal with elements s_i, z_i, w_i, p_i, v_i and q_i respectively, e is the vector of all ones. $\nabla f(x)$ is the gradient of the objective function $f(x)$. $\nabla g(x)$ and $\nabla h(x)$ are the Jacobian matrices of the constraints vectors $g(x)$ and $h(x)$ respectively.

Note that the above optimality conditions (known as KKT conditions) contain the constraints of the original problem, an equality resembling of the dual problem, as well as three equalities relating the primal slack variables to the dual variables. If μ is set to zero in (2.4), then we see that the last three equations are exactly the complementarity conditions. Usually, these last three equations are called $\mu-$*complementarity* conditions. The KKT optimality conditions are a nonlinear algebraic system with $5n + 2m + me$ equations and a similar number of unknowns, which are parameterized by the barrier parameter μ. The basis of the numerical algorithm for solving the nonlinear system (2.4) is Newton's method, which is known to be very efficient for linear and convex quadratic programming. Assuming that the system (2.4) has a solution, then for each $\mu > 0$ we get a solution $(x_\mu, s_\mu, w_\mu, v_\mu, y_\mu, z_\mu, p_\mu, q_\mu)$. The path $\{(x_\mu, s_\mu, w_\mu, v_\mu, y_\mu, z_\mu, p_\mu, q_\mu) : \mu > 0\}$ is called the *primal-dual central path*. This path plays a fundamental role in interior-point methods for mathematical programming. The interior-point method which will be presented in the next Sections is an iterative procedure that attempts with each iteration to move towards a point on the central path that is closer to the optimal point than the current point.

If the barrier parameter μ is set to zero, the KKT optimality conditions (2.4) can be partitioned into two classes as:

$$CO(t) = \begin{bmatrix} \nabla f(x) - \nabla g(x)^T y - \nabla h(x)^T z - p + q \\ g(x) \\ h(x) - s \\ x - w - l \\ x + v - u \end{bmatrix} \tag{2.5}$$

and

$$CT(s, z, w, p, v, q) = \begin{bmatrix} SZe \\ WPe \\ VQe \end{bmatrix} \tag{2.6}$$

The relations (2.5) contain the primal and the dual constraints of the problem (1.1). On the other hand, the relations (2.6) contain the complementarity conditions. For this reason, the last three equations in (2.4) are called the $\mu-$*complementarity conditions*. It is quite clear that if we could have a point $t = [x, y, z, p, q, s, w, v]$ satisfying the relation:

$$F(t) = \begin{bmatrix} CO(t) \\ CT(s, z, w, p, v, q) \end{bmatrix} = 0 \tag{2.7}$$

then the component x of this point would be the solution of the problem (1.1). For solving the system (2.7) the Newton's method will be considered. Starting with an initial point, this implies the determination of a direction of moving and a step length along this direction. The next Sections are dedicated to this subject.

The separation of the optimality conditions into these two classes is crucial for the elaboration of an efficient algorithm for (1.1), and for proving the convergence of this algorithm.

Remark For linear programming problems the system (2.4) is much simpler. The only nonlinear expressions in (2.4) are simple multiplications of slack and dual variables, and the presence of these simple nonlinearities make the subject of linear programming nontrivial. Moreover, for the linear programming case, if the primal and the dual feasible regions have a nonempty interior, then there exists a critical point for the corresponding optimality conditions (See, for example, Vanderbei [1996].) For the nonlinear programming case, it is more complicated to prove the existence of a solution for the system (2.4).

3. The Newton System

As above, denoting by $t = [x, y, z, p, q, s, w, v]^T$ then the Newton method applied to the system (2.7) will consist of the determination of the *direction* $\Delta t = [\Delta x \ \Delta y \ \Delta z \ \Delta p \ \Delta q \ \Delta s \ \Delta w \ \Delta v]^T$ as solution for the following perturbed linear algebraic system:

$$F'(t)\Delta t = -F(t) + \mu \hat{e}, \tag{3.1}$$

where $\hat{e} \in R^{5n+2m+me}$ with zero components except for the last $2n + m$ ones which are all equal to one, $F'(t)$ is the Jacobian of the function $F(t)$ computed at the current point t, and the computation of the new point

$$t^+ = t + \alpha \Delta t, \tag{3.2}$$

where parameter α is the *size of the step length*.

With this algorithm, two problems are critical : the choice of α at each iteration and the possible modification of the system (3.1) in order to find a local minimizer of (1.1). For linear and convex quadratic programming , the modification of the linear system (3.1) is never asked for, but for situations of numerical difficulties, and the step length at each iteration is computed by a simple ratio test [Lustig, Marsten and Shanno, 1990, 1991, 1992, 1994], [Vanderbei, 1990, 1994, 1996,], [Ye, 1997],[Wright, 1998]. For general convex nonlinear programming problems, it is well known that for a poor initial estimate, Newton's method may diverge, so once again the linear system (3.1) needs not be modified, but the technique for choosing the value of parameter α at each iteration is more complex. In order to achieve convergence to a solution of the system (2.4), El-Bakry, Tapia, Tsuchiya and Zhang [1996] introduced a merit function and showed that for a proper choice of the barrier parameter μ, there existed an α at each iteration such that the algorithm (3.2) converged to a solution of (1.1), provided that the Jacobian $F'(t)$ in (3.1) remained nonsingular. An implementation of this variant of the algorithm was tested by Shanno, Breitfeld and Simantiraki [1996], and Shanno and Simantiraki [1997]. On the Hoch and Schittkowski [1981] set of test problems they found that while the algorithm had often been efficient, giving the optimal solution, it had also converged to local maxima or saddle points, all of which satisfied the necessary first order optimality conditions. Besides, for some problems, the Jacobian $F'(t)$ sometimes becomes singular, causing troubles to the algorithm. A modification of the merit function of El-Bakry *et al* has been considered by Vanderbei and Shanno [1997] where the l_2 norm of the constraints, multiplied by a parameter β, is introduced. They proved that there was a value of β such that the direction given by the Newton system be a descent direction for this merit function. The corresponding algorithm was implemented as a nonlinear version of LOQO [Vanderbei, 1994] and tested on the Hoch and Schittkowski set of problems, eight large-scale nonlinear programming problems, and the Mittelmann quadratic programming set of problems [Mittelmann 1997]. Comparisons with MINOS [Murtagh and Saunders, 1995] and LANCELOT [Conn, Gould and Toint, 1992] showed how robust and efficient this interior point approach proved to be.

4. The Newton Direction Determination

The Newton system for the direction determination involves the computation of the Jacobian matrix of the function $F(t)$ in (2.7). From (2.4) we get the following linear algebraic system:

$$\begin{bmatrix} K(x,y,z) & -G(x)^T & -H(x)^T & -I & I & & & & \\ G(x) & & & & & & & & \\ H(x) & & & & & -I & & & \\ I & & & & & & -I & & \\ I & & & & & & & I & \\ & & S & & & Z & & & \\ & & & W & & & P & & \\ & & & & V & & & Q \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta p \\ \Delta q \\ \Delta s \\ \Delta w \\ \Delta v \end{bmatrix} = \begin{bmatrix} \chi \\ -g(x) \\ -h(x)+s \\ l+w-x \\ u-v-x \\ -SZe+\mu e \\ -WPe+\mu e \\ -VQe+\mu e \end{bmatrix} \tag{4.1}$$

This system is not symmetrical, but it is easy to symmeterize by multiplying the second and the third equations by -1, and the last three equations by S^{-1}, W^{-1} and V^{-1} respectively. After completing this very simple algebraic operation, and after eliminating variables Δs, Δw and Δv as:

$$\begin{aligned} \Delta s &= \mu Z^{-1}e - Se - Z^{-1}S\Delta z, \\ \Delta w &= \mu P^{-1}e - We - P^{-1}W\Delta p, \\ \Delta v &= \mu Q^{-1}e - Ve - Q^{-1}V\Delta q, \end{aligned} \tag{4.2}$$

the following *reduced Newton system* is obtained:

$$\begin{bmatrix} K(x,y,z) & -G(x)^T & -H(x)^T & -I & I \\ -G(x) & & & & \\ -H(x) & & -Z^{-1}S & & \\ -I & & & -P^{-1}W & \\ I & & & & -Q^{-1}V \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta p \\ \Delta q \end{bmatrix} = \begin{bmatrix} \chi \\ g(x) \\ h(x) - \mu Z^{-1}e \\ x - l - \mu P^{-1}e \\ u - x - \mu Q^{-1}e \end{bmatrix} \quad (4.3)$$

where

$$K(x,y,z) = \nabla^2 f(x) - \sum_{i=1}^{me} y_i \nabla^2 g_i(x) - \sum_{i=1}^{m} z_i \nabla^2 h_i(x), \quad (4.4)$$

$$\begin{aligned} G(x) &= \nabla g(x), \\ H(x) &= \nabla h(x), \\ \chi &= -\nabla f(x) + G(x)^T y + H(x)^T z + p - q. \end{aligned} \quad (4.5)$$

Notice that the matrix of the reduced Newton system (4.3) is symmetrical, undefined and $(3n + m + me)-$dimensional. Solving this system, or a reduction of it, together with (4.2) we get the direction $\Delta t = [\Delta x \ \Delta y \ \Delta z \ \Delta p \ \Delta q \ \Delta s \ \Delta w \ \Delta v]^T$. The existence of a solution for (4.3) implies the existence of the inverse of the diagonal matrices Z, P and Q,thus assumng that a condition of initialization of the algorithm is satisfied.

The standard Newton method assumptions for the problem (1.1) are as follows [Dennis and Schnabel, 1983]:

(A1) There exists a solution for the problem (1.1) and the associated dual variables, satisfying the KKT conditions given by (2.4).

(A2) The Hessian matrices $\nabla^2 f(x)$, $\nabla^2 g_i(x)$, $\nabla^2 h_i(x)$ for all i exist and are locally Lipschitz continuous at x^*.

(A3) The set of vectors $\{\nabla g_1(x^*), ..., \nabla g_{me}(x^*)\} \cup \{\nabla h_i(x^*) : i \in A(x^*)\}$ is linearly independent, where $A(x^*)$ is the set of inequality constraints active in x^*.

(A4) For every vector $d \neq 0$ satisfying $\nabla g_i(x^*)^T d = 0$, $i = 1, ..., me$ and $\nabla h_i(x^*)^T d = 0$, $i \in A(x^*)$ we have $d^T K(x^*, y^*, z^*) d > 0$.

(A5) For $i = 1, ..., m$, $z_i h_i(x^*) > 0$, and for $j = 1, ..., n$, $p_j(x_j^* - l_j) > 0$, $q_j^*(u_j - x_j^*) > 0$.

Proposition 1 *Suppose that the conditions (A1)-(A5) are satisfied and* $s^* = h(x^*)$, $w^* = x^* - l$, $v^* = u - x^*$, *then the Jacobian matrix* $F'(t^*)$ *of* $F(t)$ *given by the matrix from (4.1) is nonsingular.*
*Proof*This follows immediately. Let the reduced problem be, in which only the inequality constraints active in the optimum point are considered. Then, from the theory of equality constrained optimization, it follows that the matricial block given by the first five rows and columns of the matrix from (4.1) is nonsingular. Hence, the nonsingularity of (4.1) follows from the strict complementarity condition (A5) and the nonsingularity of the matricial block from (4.1). □

Generally, in current implementations, the reduced Newton system (4.3) is not considered in the form in which it appears. Usually, we continue to reduce it by simple algebraic manipulations. Considering (4.3) and eliminating the variables Δz, Δp and Δq as:

$$\begin{aligned} \Delta z &= \mu S^{-1}e - S^{-1}Zh(x) - S^{-1}ZH(x)\Delta x, \\ \Delta p &= \mu W^{-1}e - W^{-1}P(x - l) - W^{-1}P\Delta x, \\ \Delta q &= \mu V^{-1}e - V^{-1}Q(u - x) + V^{-1}Q\Delta x, \end{aligned} \quad (4.6)$$

the system (4.3) gets reduced to the following linear symmetric system:

$$\begin{bmatrix} \bar{K} & -G(x)^T \\ G(x) & \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r + \mu \bar{r} \\ g(x) \end{bmatrix}, \tag{4.7}$$

where

$$\begin{aligned} \bar{K} &= K(x,y,z) + H(x)^T S^{-1} Z H(x) + W^{-1} P + V^{-1} Q, \\ r &= \chi - H(x)^T S^{-1} Z h(x) - W^{-1} P(x-l) + V^{-1} Q(u-x), \\ \bar{r} &= H(x)^T S^{-1} e + W^{-1} e - V^{-1} e. \end{aligned} \tag{4.8}$$

Solving the system (4.7) we get the variables Δx and Δy which from (4.6) and (4.2) and the rest of the system variables, (4.1) can be computed with. Notice that, as in the case of linear programming, we must solve a symmetric, undefined, algebraic system, but in this case, the Jacobian and Hessian matrices of the functions of the problems being involved, this is much more complicated.

5. The Merit Function

Solving the system (4.7), and considering (4.6) and (4.2) respectively, we get a direction Δt pointing to the central path. The problem we must solve now is to determine the step length along this direction. This is achieved by means of a merit function associated with the optimality conditions (2.4). The idea of a merit function is to enable progress towards a local minimum of the problem by conserving the feasibility of the solution. The merit function used for the line search is the squared l_2-norm of the KKT conditions (2.4), i.e.

$$\Phi(t) = \| F(t) \|_2^2 = F(t)^T F(t). \tag{5.1}$$

Denote by $\Phi_k = \Phi(t_k)$ the value of the merit function at the iteration t_k, and $\Phi_k(\alpha) = \Phi(t_k + \alpha \Delta t_k)$ which illustrates the dependence of the merit function on the step length α. Clearly, $\Phi_k(0) = \Phi(t_k) = \Phi_k$. Hence

$$\Phi_k(\alpha) = F(t_k + \alpha \Delta t_k)^T F(t_k + \alpha \Delta t_k). \tag{5.2}$$

The following proposition shows that the direction given by the perturbed Newton's system (3.1):

$$\Delta t_k = \left[F^{'}(t_k) \right]^{-1} \left[-F(t_k) + \mu \hat{e} \right] \tag{5.3}$$

is a descent direction for the merit function $\Phi(t)$.

Proposition 2 *The direction Δt_k given by (5.3), the solution of the perturbed Newton system (3.1) is a descent direction for the merit function (5.1).*

Proof Considering the derivative of $\Phi_k(\alpha)$ at $\alpha = 0$ we get:

$$\begin{gathered} \Phi^{'}(0) = 2F(t_k)^T F^{'}(t_k) \Delta t_k = \\ 2F(t_k)^T F^{'}(t_k) \left[F^{'}(t_k) \right]^{-1} \left[-F(t_k) + \mu \hat{e} \right] = \\ 2F(t_k)^T \left[-F(t_k) + \mu \hat{e} \right] = \\ -2 \| F(t_k) \|_2^2 + 2\mu F(t_k)^T \hat{e}, \end{gathered}$$

hence $\Phi^{'}(0) < 0$ if and only if $2\mu F(t_k)^T \hat{e} \leq 2 \| F(t_k) \|_2^2$. This suggests the following estimation of the barrier parameter used in the logarithmic barrier function (2.2):

$$\mu \leq \frac{\|F(t_k)\|_2^2}{F(t_k)^T \hat{e}} = \frac{\|F(t_k)\|_2^2}{s^T z + w^T p + v^T q}. \tag{5.4}$$

So, at every iteration, choosing the value of the barrier parameter μ as in (5.4), the direction Δt_k given by (5.3) is indeed a descent direction for the merit function $\Phi(t_k)$.□

The relation (5.4) is a very important estimation of the barrier parameter μ. We will see that the numerical experiments recommend a much smaller value for it.

6. The Step Length Determination

The step length computation in algorithm (3.2) is a very important ingredient. For the unconstrained optimization the value of α is determined by means of the *Wolfe conditions* [Wolfe, 1969, 1971], [Zoutendijk, 1970], [Dennis and Schnabel, 1983] , [Nocedal, 1992]. For linear programming or linear constrained optimization the step length is determined by a simple ratio test. For nonlinear programming the strategy for choosing α at each iteration becomes more complex. In such a case it is necessary to impose some additional constraints for making sure that the current iterations are generated as close as possible to the central path.

As we have learned from linear programming, also for the problem (1.1), the distance from centrality is given by:

$$\xi_k = \frac{\min\{s_i z_i,\ w_i p_i,\ v_i q_i\}}{\frac{s^T z + w^T p + v^T q}{2n+m}}. \tag{6.1}$$

Clearly, $0 < \xi_k \leq 1$, and $\xi_k = 1$ if and only if $s_i z_i$, $w_i p_i$, $v_i q_i$ are equal to a constant for all values of i. To specify a value of α, firstly the following function is introduced:

$$\Theta^I(\alpha) = \frac{\min\{s_i(\alpha)z_i(\alpha),\ w_i(\alpha)p_i(\alpha),\ v_i(\alpha)q_i(\alpha)\}}{\frac{s^T(\alpha)z(\alpha)+w^T(\alpha)p(\alpha)+v^T(\alpha)q(\alpha)}{2n+m}} - \gamma\tau_1 \tag{6.2}$$

where τ_1 is the initial distance from centrality:

$$\tau_1 = \frac{\min\{s_i^0 z_i^0,\ w_i^0 p_i^0,\ v_i^0 q_i^0\}}{\frac{s^{0T} z^0 + w^{0T} p^0 + v^{0T} q^0}{2n+m}}, \tag{6.3}$$

and $\gamma \in (0,1)$ is a constant thereby we can modify the distance from centrality. Notice that for $t = t_0$ and $\gamma = 1$, it follows that $\Theta^I(0) = 0$. In addition, $\Theta^I(\alpha)$ is a piecewise quadratic function. To choose the step length α_k at each iteration it is necessary that α_k satisfies $\Theta^I(\alpha) \geq 0$ for all $\alpha \in [0, \alpha_k]$, i.e.

$$\frac{\min\{s_i(\alpha)z_i(\alpha),\ w_i(\alpha)p_i(\alpha),\ v_i(\alpha)q_i(\alpha)\}}{\frac{s^T(\alpha)z(\alpha)+w^T(\alpha)p(\alpha)+v^T(\alpha)q(\alpha)}{2n+m}} \geq \gamma_k\tau_1 \tag{6.4}$$

where the variables of the problem are considered at the $k-$th iteration and $\gamma_k \in (0,1)$. Since $\Theta^I(\alpha)$ is a piecewise quadratic function, from (6.4) it follows that α_k can be easily computed.

Considering now the merit function (5.1), Wolfe's conditions for its minimization are:

$$\Phi(t_{k+1}) \leq \Phi(t_k) + \beta\alpha_k \nabla\Phi(t_k)^T \Delta t_k, \tag{6.5}$$

$$\nabla\Phi(t_{k+1})^T \Delta t_k \geq \delta\nabla\Phi(t_k)^T \Delta t_k, \tag{6.6}$$

where $0 < \beta < \delta < 1$ are parameters fixing the reduction of the merit function, as well as the rate of decreasing this function along the direction Δt_k. Having in view that $\Phi(t_{k+1}) = \Phi_k(\alpha)$, the first Wolfe's condition (6.5) is equivalent to:

$$\Phi_k(\alpha) \leq \Phi_k(0) + \beta\alpha_k\Phi_k'(0). \tag{6.7}$$

Proposition 3 *For the merit function (5.1), taking*

$$\mu_k = \sigma_k \frac{s_k^T z_k + w_k^T p_k + v_k^T q_k}{2n+m} \tag{6.8}$$

where $\sigma_k \in (0,1)$ we have:

$$\Phi_k'(0) = -2\left[\Phi_k(0) - \frac{\sigma}{2n+m}(s^T z + w^T p + v^T q)^2\right]. \tag{6.9}$$

Proof By direct computation we get:

$$\Phi_k'(0) = -2\Phi_k(0) + 2\mu F(t_k)^T \hat{e} = -2\Phi_k(0) + 2\mu[s^T z + w^T p + v^T q] =$$
$$-2\Phi_k(0) + 2\sigma\frac{s^T z + w^T p + v^T q}{2n+m}[s^T z + w^T p + v^T q] =$$
$$-2\left[\Phi_k(0) - \frac{\sigma}{2n+m}(s^T z + w^T p + v^T q)^2\right]. \square$$

Proposition 4 *At every iteration:*

$$\frac{(s^T z + w^T p + v^T q)^2}{2n+m} \leq \Phi_k(0). \tag{6.10}$$

Proof By simple algebraic manipulation we get:

$$\frac{(s^T z + w^T p + v^T q)^2}{2n+m} \leq \| SZe \|_2^2 + \| WPe \|_2^2 + \| VQe \|_2^2 \leq$$
$$\| SZe \|_2^2 + \| WPe \|_2^2 + \| VQe \|_2^2 + \| CO(t_k) \|_2^2 = \Phi_k(0). \square$$

It is easy to show that the estimation of the barrier parameter μ_k given by (6.8) with $\sigma_k \in (0,1)$, is smaller than the estimation recommended by (5.4). Hence, the μ_k given by (6.8) ensures the descent character of the direction Δt_k. Moreover, the following proposition gives an estimation of the reduction of the values of the merit function.

Proposition 5 *The direction Δt_k solution of the perturbed Newton system (3.1) with μ given by (6.8) is a descent direction for the merit function $\Phi(t)$ at every t_k. Moreover, if the first Wolfe condition (6.5) is satisfied, then*

$$\Phi_k(\alpha_k) \leq [1 - 2\alpha_k\beta(1-\sigma_k)]\Phi_k(0). \tag{6.11}$$

Proof As we know $\Phi_k'(0) = -2 \| F(t_k) \|_2^2 + 2\mu_k(s^T z + w^T p + v^T q)$. Considering μ_k as in (6.8) it follows that:

$$\Phi_k'(0) = -2\Phi_k(0) + 2\sigma_k\frac{(s^T z + w^T p + v^T q)^2}{2n+m} \leq$$
$$-2\Phi_k(0) + 2\sigma_k\Phi_k(0) = -2\Phi_k(0)(1-\sigma_k) < 0,$$

proving the descent character of the direction given by the perturbed Newton system. Moreover, from (6.7) and taking into account the above propositions (3 and 4) we have:

$$\Phi_k(\alpha) \leq \Phi_k(0) + \beta\alpha_k\Phi_k'(0) =$$
$$\Phi_k(0) + \beta\alpha_k\left[-2\left(\Phi_k(0) - \sigma_k\frac{(s^T z + w^T p + v^T q)^2}{2n+m}\right)\right] \leq$$
$$\Phi_k(0) - 2\beta\alpha_k\Phi_k(0) + 2\beta\alpha_k\sigma_k\Phi_k(0) =$$
$$[1 - 2\alpha_k\beta(1-\sigma_k)]\,\Phi_k(0),$$

which proves the proposition. $\square$

This proposition shows that the sequence $\{\Phi_k\}$ is monotonous and nonincreasing, therefore for all $k : \Phi_k \leq \Phi_0$. Moreover, if the sequence of step length $\{\alpha_k\}$ is bounded away from zero and the parameter σ_k is bounded away from one at every iteration, it follows that the merit function is linearly convergent to zero. Hence the above (6.11) inequality is equivalent to:

$$\frac{\|F(t_{k+1})\|_2}{\|F(t_k)\|_2} \leq [1 - 2\alpha_k\beta(1-\sigma_k)]^{1/2}. \tag{6.12}$$

Some numerical examples illustrate that a problem that may lead to the non-convergence of the algorithm is that of the sequence $\{\| CT(s_k, z_k, w_k, p_k, v_k, q_k) \|\}$ converging to zero faster than the sequence $\{\Phi(t_k)\}$. In such cases the sequence of the step-lengths $\{\alpha_k\}$ is decreasing to zero, thus determining the nonconvergence of the algorithm. In order to avoid this situation, let us introduce the following function:

$$\begin{aligned}\Theta^{II}(\alpha) = s(\alpha)^T z(\alpha) + w(\alpha)^T p(\alpha) + v(\alpha)^T q(\alpha) - \\ \gamma\tau_2 \| CO(t(\alpha)) \|_2\end{aligned} \tag{6.13}$$

where

$$\tau_2 = \frac{s_0^T z_0 + w_0^T p_0 + v_0^T q_0}{\|CO(t_0)\|_2}, \tag{6.14}$$

and $\gamma \in (0,1)$ is a constant, the same as in (6.2). Notice that for $t = t_0$ and $\gamma = 1$; $\Theta^{II}(0) = 0$. $\Theta^{II}(\alpha)$ is generally nonlinear. For choosing the step-length α_k at every iteration, it is necessary that α_k satisfies :

$$\Theta^{II}(\alpha_k) \geq 0. \tag{6.15}$$

Proposition 6 *Let $\{t_k\}$ be a sequence generated by (3.1). Then*

$$\min\{1, 0.5\tau_2\}\Phi(t_k) \leq (s_k^T z_k + w_k^T p_k + v_k^T p_k)^2 \leq (2n+m)\Phi(t_k). \tag{6.16}$$

Proof The second inequality follows from Proposition 4. So, we will have to prove only the first one. Since $\Theta^i(\alpha_k) \geq 0$, for $i = 1, 2$, from (6.13) with $\gamma_k \geq 1/2$, we have:

$$(s_k^T z_k + w_k^T p_k + v_k^T p_k) \geq (1/2)\tau_2 \| CO(t_k) \|_2 .$$

It follows that

$$\begin{aligned}(s_k^T z_k + w_k^T p_k + v_k^T p_k) \geq \\ \tfrac{1}{2} [\| SZe \|_2 + \| WPe \|_2 + \| VQe \|_2 + 0.5\tau_2 \| CO(t_k) \|_2] \geq \\ \tfrac{1}{2} \min\{1, 0.5\tau_2\} \| F(t_k) \|_2,\end{aligned}$$

which completes the proof. □

This proposition shows that the transversality conditions are bounded, and γ_k must be selected as a decreasing sequence with $1/2 \leq \gamma_k \leq \gamma_{k-1}$. Having in view all these developments, at each iteration, the step length α_k is computed as a solution of the following system of algebraic inequalities:

$$\begin{array}{ll} a) & l \leq x_k + \alpha_k \Delta x_k \leq u, \\ b) & s(\alpha_k), z(\alpha_k), w(\alpha_k), p(\alpha_k), v(\alpha_k), q(\alpha_k) > 0, \\ c) & \Theta^I(\alpha) \geq 0, \alpha \in [0, \alpha_k], \\ d) & \Theta^{II}(\alpha_k) \geq 0, \\ e) & \Phi_k(\alpha_k) \leq \Phi_k(0) + \alpha_k\beta\Phi_k'(0), \\ f) & \nabla\Phi(\alpha_k)^T \Delta t_k \geq \delta\nabla\Phi(0)^T \Delta t_k. \end{array} \tag{6.17}$$

where $0 < \beta < \delta < 1$.

In order to determine a value α_k for the step length satisfying (6.17) we shall consider a strategy of *interval reducing*. Clearly, the first two relations (6.17a) and (6.17b) are very simple to implement. The same as in the linear programming case the corresponding ratio test is performed, thus obtaining a value α_m which maintains the positivity of the variables as well as simple bounds on the variables of the problem. Then a value $\alpha_k \in (0, \alpha_m]$ which satisfies the conditions (6.17c - 6.17f) is selected.

7. The Primal-Dual Interior-Point Algorithm

The interior point algorithm for solving (1.1), based on the above developments, has three main parts which refer to the computing of the direction of moving, the barrier parameter, and the step length.

Algorithm PCNC

Step 1 *Initialization.* Choose the initial point $t_0 = [x_0\ y_0\ z_0\ p_0\ q_0\ s_0\ w_0\ v_0]$ such that $l \le x_0 \le u$, $(s_0, z_0) > 0$, $(w_0, p_0) > 0$, $(v_0, q_0) > 0$, as well as the values of the parameters: $\beta \in (0, 1/2]$, $\gamma_{k-1} = 1$, and $\rho \in (0, 1)$. Set $k = 0$.

Step 2 *Test of convergence.* Compute the value of the merit function: $\Phi(t_k) = F(t_k)^T F(t_k)$. If $\Phi(t_k) \le \varepsilon$, stop, else continue with Step 3.

Step 3 *Evaluation of the barrier parameter.* Choose $\sigma_k \in (0, 1)$. Using (6.8) compute the value of the barrier parameter μ_k.

Step 4 *Computing the direction of moving.* Compute the perturbed Newton direction Δt_k by solving the reduced Newton system (4.7) and using the relations (4.6) and (4.2).

Step 5 *Step length determination.*

a) Choose $\frac{1}{2} \le \gamma_k \le \gamma_{k-1}$.

b) Compute α_m as the maximum value of α_k which satisfies the conditions (6.17a) and (6.17b).

c) Determine $\alpha_k^I \in (0, \alpha_m]$ as the smallest positive root such that $\Theta^I(\alpha) \ge 0$ for all $\alpha \in (0, \alpha_k^I]$.

d) Determine $\alpha_k^{II} \in (0, \alpha_k^I]$ such that $\Theta^{II}(\alpha_k^{II}) \ge 0$.

e) Set $\bar{\alpha}_k = \min(\alpha_k^I, \alpha_k^{II})$.

f) Let $\alpha_k = \rho^j\ \bar{\alpha}_k$, where j is the smallest non-negative integer such that α_k satisfies the condition (6.17e)

Step 6 *Updating of the variables.* Set $t_{k+1} = t_k + \alpha_k \Delta t_k$, $k = k + 1$ and go to Step 2.

Some remarks are in order:

1. *The initial point.* Generally, nonlinear programming requires that an initial starting point be given as part of the problem data. For an interior-point algorithm, besides the specification of the initial values for x_j, the initial values for variables y_i, z_i, p_i, etc. must be determined by the program. For slack variables w_i and v_i we can simply consider the initial values as: $w_i = x_i - l_i$ and $v_i = u_i - x_i$, for $i = 1, ..., n$, respectively. For slack variables s_i we can compute $h(x_0)$ and set $s_i = h_i(x_0)$, for $i = 1, ..., m$. There are two problems with this approach for computing the initial values for the s_i variables. Firstly, if x_0 is not feasible, then s_i computed as above gives an initial negative value to some components. Secondly, even if x_0 is feasible, it may lie on the boundary of the feasible region of the problem or so close to it that some initial values of s_i are very close to zero, thus precluding the algorithm from making any progress. To overcome this difficulty, as Vanderbei and Shanno [1997] suggested, it is necessary to specify a value for a parameter $\vartheta > 0$ so that all initial values of variables constrained to be non negative are at least as large as ϑ. Hence s_i is initially set as: $s_i = \max\{h_i(x_0), \vartheta\}$. (In our tests we have considered $\vartheta = 1$.) The initial values of the variables y_i, $i = 1, ..., me$, are computed as a solution for the following linear system:

$$[\nabla g(x_0) \nabla g(x_0)^T] y = \nabla g(x_0)[\nabla f(x_0) - \nabla h(x_0)^T z_0 - p_0 + q_0]. \tag{7.1}$$

Proposition 7 *Suppose that the functions of the problem $f(x)$, $g(x)$ and $h(x)$ are twice continuously differentiable and the derivative of $CO(t)$ given by (2.5) is Lipschitz continuous. If the columns of $\nabla g(x)$ are linearly independent, then the variable y given by (7.1) is well defined and bounded.*

Proof From the first relation of (2.5) it follows that:

$$y = [\nabla g(x) \nabla g(x)^T]^{-1} \nabla g(x)[\nabla f(x) - \nabla h(x)^T z - p + q]. \tag{7.2}$$

Having in view that x is in a compact set and the Jacobian $\nabla g(x)$ is of full rank on rows, then the inverse of the matrix in (7.2) exists, i.e. y is well defined. The boundness of y follows from the conditions of the proposition. □

2. *The search direction.* In Proposition 2 above we showed that the step direction Δt given by (5.3)

has desirable descent properties for the merit function provided that the value of the barrier parameter is selected as in (5.4).

To determine the search direction the reduced Newton system (4.7) is solved by the Cholesky factorization. After having determined Δx and Δy by using (4.6) and (4.2) the rest of variables are very easy to compute.

Proposition 8 *Suppose that the functions of the problem $f(x)$, $g(x)$ and $h(x)$ are twice continuously differentiable, the derivative of $CO(t)$ given by (2.5) is Lipschitz continuous and the set of gradients $\{\nabla g_1(x_k), ..., \nabla g_{me}(x_k)\} \cup \{\nabla h_i(x_k),\ i \in A(x^*)\}$ is linearly independent for k sufficiently large, where $A(x^*)$ is the set of active inequality constraints in x^*. Then the sequence of search directions $\{\Delta t_k\}$ generated by the algorithm PCNC is bounded.*

Proof Symmeterizing the matrix $F^{'}(t_k)$ and rearranging the order of its rows and columns, we have

$$F^{'}(t_k) = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$$

where

$$A = \begin{bmatrix} K(x_k, y_k, z_k) & -G(x_k)^T & -H(x_k)^T \\ -G(x_k) & & \\ -H(x_k) & & \end{bmatrix}$$

and the matrices B and C can be very easily identified from $F^{'}(t)$. By assumption of the proposition, the matrix A is invertible and $\| A^{-1} \|$ is uniformly bounded. A straightforward computation shows that:

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix}^{-1} =$$

$$\begin{bmatrix} A^{-1} + A^{-1}B(C - B^T A^{-1}B)^{-1}B^T A^{-1} & -A^{-1}B(C - B^T A^{-1}B)^{-1} \\ -(C - B^T A^{-1}B)^{-1}B^T A^{-1} & (C - B^T A^{-1}B)^{-1} \end{bmatrix}$$

which is bounded, since every matrix involved is bounded. Hence $\left[F^{'}(t_k)\right]^{-1}$ is uniformly bounded, which proves the Proposition. □

3. *Measures of progress.* We assume that the functions of the problem $f(x)$, $g(x)$ and $h(x)$ are twice continuously differentiable and the derivative of $CO(t)$ in (2.5) is Lipschitz continuous. From (6.17a) we notice that the iteration sequence $\{x_k\}$ is bounded. Moreover, the descent character of the Newton direction and the specific choice of the step length α ensure that $\Delta t \to 0$ along the iterations. The following propositions show the progress of the algorithm in one iteration regarding the primal and the dual feasibility as well as the complementarity.

Proposition 9 *Let us define:*

$$\begin{aligned} e_1(x, s) &= s - h(x), \\ e_2(x, w) &= l + w - x, \\ e_3(x, v) &= u - v - x. \end{aligned} \tag{7.3}$$

Then

$$\begin{aligned} e_1(\bar{x}, \bar{s}) &= (1 - \alpha)e_1(x, s) + o(\alpha) \\ e_2(\bar{x}, \bar{w}) &= (1 - \alpha)e_2(x, w) \\ e_3(\bar{x}, \bar{v}) &= (1 - \alpha)e_3(x, v) \end{aligned} \tag{7.4}$$

Proof By direct computation we have:

$$\begin{aligned}e_1(\bar{x},\bar{s}) = \bar{s} - h(\bar{x}) = s + \alpha\Delta s - h(x+\alpha\Delta x) = \\ s + \alpha\Delta s - h(x) - \alpha\nabla h(x)\Delta x + o(\alpha) = \\ s - h(x) + \alpha[\Delta s - \nabla h(x)\Delta x] + o(\alpha) = \\ (1-\alpha)e_1(x,s) + o(\alpha),\end{aligned}$$

where the last equality follows from the third block of equations in (4.1). The analysis goes similar for e_2 and e_3.□

Proposition 10 *Define $\bar{\chi}$ as the value of χ from (4.5) computed in $\bar{t} = t + \alpha\Delta t$. Then*

$$\bar{\chi} \leq (1-\alpha)\chi. \tag{7.5}$$

Proof By direct computation we get:

$$\begin{aligned}\bar{\chi} = -\nabla f(\bar{x}) + G(\bar{x})^T\bar{y} + H(\bar{x})^T\bar{z} + \bar{p} - \bar{q} = \\ \chi + \alpha[-K(x,y,z)\Delta x + G(x)^T\Delta y + H(x)^T\Delta z + \Delta p - \Delta q] + \\ \alpha^2[\sum_{i=1}^{me}\Delta y_i\nabla^2 g_i(x) + \sum_{i=1}^{m}\Delta z_i\nabla^2 h_i(x)]\Delta x.\end{aligned}$$

Using the first equality from (4.1) it follows that

$$\bar{\chi} = (1-\alpha)\chi + \alpha^2[\sum_{i=1}^{me}\Delta y_i\nabla^2 g_i(x) + \sum_{i=1}^{m}\Delta z_i\nabla^2 h_i(x)]\Delta x.$$

Now having in view that the derivative of $CO(t)$ from (2.5) is Lipschitz continuous it follows that $\bar{\chi} \leq (1-\alpha)\chi$. □

Proposition 11 *Assume that $\| s \|_\infty, \| w \|_\infty, \| v \|_\infty$ are bounded by a large real number M, and α defined by (6.17) is such that $\alpha \leq 1$. Let us define:*

$$\omega = s^T z + w^T p + v^T q. \tag{7.6}$$

Then

$$\bar{\omega} \leq \omega[1 - \alpha(1-\sigma)] + M[\| \Delta z \|_1 + \| \Delta p \|_1 + \| \Delta q \|_1] \tag{7.7}$$

Proof By simple algebraic manipulations we get:

$$\begin{aligned}\bar{\omega} = \bar{s}^T\bar{z} + \bar{w}^T\bar{p} + \bar{v}^T\bar{q} = \\ (s+\alpha\Delta s)^T(z+\alpha\Delta z) + (w+\alpha\Delta w)^T(p+\alpha\Delta p) + (v+\alpha\Delta v)^T(q+\alpha\Delta q) = \\ \omega + \alpha[(s^T\Delta z + z^T\Delta s) + (w^T\Delta p + p^T\Delta w) + (v^T\Delta q + q^T\Delta v)] + \\ \alpha^2[\Delta s^T\Delta z + \Delta w^T\Delta p + \Delta v^T\Delta q].\end{aligned}$$

But,

$$\begin{aligned}s^T\Delta z + z^T\Delta s = e^T(S\Delta z + Z\Delta s) = e^T(\mu e - SZe) = \mu m - s^T z, \\ w^T\Delta p + p^T\Delta w = e^T(W\Delta p + P\Delta w) = e^T(\mu e - WPe) = \mu n - w^T p, \\ v^T\Delta q + q^T\Delta v = e^T(V\Delta q + Q\Delta v) = e^T(\mu e - VQe) = \mu n - v^T q,\end{aligned}$$

where the last two equalities are from (4.1). So,

$$\bar{\omega} = \omega + \alpha[(m+2n)\mu - \omega] + \alpha^2[\Delta s^T\Delta z + \Delta w^T\Delta p + \Delta v^T\Delta q],$$

and from (6.8) it follows that:

$$\bar{\omega} = \omega[1-\alpha(1-\sigma)] + \alpha^2[\Delta s^T \Delta z + \Delta w^T \Delta p + \Delta v^T \Delta q].$$

Now we shall estimate the term in α^2 as follows:

$$\alpha^2 \mid \Delta s^T \Delta z + \Delta w^T \Delta p + \Delta v^T \Delta q \mid \leq \alpha^2[\mid \Delta s^T \Delta z \mid + \mid \Delta w^T \Delta p \mid + \mid \Delta v^T \Delta q \mid],$$

But,

$$\alpha^2 \mid \Delta s^T \Delta z \mid = \alpha^2 \mid \sum_i \Delta s_i \Delta z_i \mid \leq \alpha^2 \sum_i \mid \Delta s_i \mid\mid \Delta z_i \mid \leq$$
$$\alpha^2 (\max_i \mid \Delta s_i \mid)(\sum_i \mid \Delta z_i \mid) = \alpha^2 \parallel \Delta s \parallel_\infty \parallel \Delta z \parallel_1 =$$
$$\alpha \parallel \alpha \Delta s \parallel_\infty \parallel \Delta z \parallel_1,$$

where the last inequality is a trivial case of Hölder's inequality. Clearly, similar estimates can be made for $\alpha^2 \mid \Delta w^T \Delta p \mid$ and $\alpha^2 \mid \Delta v^T \Delta q \mid$. Substituting these expressions into the last expression for $\bar{\omega}$, we get:

$$\bar{\omega} \leq \omega[1-\alpha(1-\sigma)] +$$
$$\alpha[\parallel \alpha \Delta s \parallel_\infty \parallel \Delta z \parallel_1 + \parallel \alpha \Delta w \parallel_\infty \parallel \Delta p \parallel_1 + \parallel \alpha \Delta v \parallel_\infty \parallel \Delta q \parallel_1].$$

Now, we use the specific choice of the step length α to get a bound on $\parallel \alpha \Delta s \parallel_\infty$, $\parallel \alpha \Delta w \parallel_\infty$ and $\parallel \alpha \Delta v \parallel_\infty$. Indeed, (6.17a and 6.17b) imply that

$$\alpha \leq \frac{s_i}{|\Delta s_i|}, \text{ for all } i = 1, ..., m, \text{ hence } \parallel \alpha \Delta s \parallel_\infty \leq \parallel s \parallel_\infty.$$

Similarly, $\parallel \alpha \Delta w \parallel_\infty \leq \parallel w \parallel_\infty$ and $\parallel \alpha \Delta v \parallel_\infty \leq \parallel v \parallel_\infty$. Having in view that $\parallel s \parallel_\infty, \parallel w \parallel_\infty$ and $\parallel v \parallel_\infty$ are bounded by the real number M, the following estimate of the complementarity is obtained:

$$\bar{\omega} \leq \omega[1-\alpha(1-\sigma)] + M[\parallel \Delta z \parallel_1 + \parallel \Delta p \parallel_1 + \parallel \Delta q \parallel_1]. \ \square$$

Another estimation of the complementarity can be made as in

Proposition 12 *Assume that* $\parallel z \parallel_\infty, \parallel p \parallel_\infty, \parallel q \parallel_\infty$ *are bounded by a large real number* M, *and* α *defined by (6.17) is such that* $\alpha \leq 1$. *Then*

$$\bar{\omega} = \leq \omega[1-\alpha(1-\sigma)] + M[\parallel e_1 \parallel_1 + \parallel e_2 \parallel_1 + \parallel e_3 \parallel_1]. \quad (7.8)$$

Proof As in Proposition 11 above we can write:

$$\bar{\omega} = \omega[1-\alpha(1-\sigma)] + \alpha^2[\Delta s^T \Delta z + \Delta w^T \Delta p + \Delta v^T \Delta q].$$

Now from (4.1) we get: $\Delta s = H(x)\Delta x - e_1$, $\Delta w = \Delta x - e_2$ and $\Delta v = e_3 - \Delta x$. Substituting these expressions of Δs, Δw and Δv into the expresion of $\bar{\omega}$ we get

$$\bar{\omega} = \omega[1-\alpha(1-\sigma)] + \alpha^2[\Delta x^T(H^T \Delta z + \Delta p - \Delta q) - e_1^T \Delta z - e_2^T \Delta p + e_3^T \Delta q].$$

But from the first equality in (4.1) $H^T \Delta z + \Delta p - \Delta q = K\Delta x - G^T \Delta y - \chi$. Hence

$$\bar{\omega} = \omega[1-\alpha(1-\sigma)] + \alpha^2[\Delta x^T K \Delta x - \Delta x^T G^T \Delta y - \Delta x^T \chi - e_1^T \Delta z - e_2^T \Delta p + e_3^T \Delta q].$$

Having in view that the functions of the problem are twice continuously differentiable and the derivative of $CO(t)$ given by (2.5) is Lipschitz continuous, it follows that

$$\bar{\omega} \leq \omega[1-\alpha(1-\sigma)] + \alpha^2[\mid e_1^T \Delta z \mid + \mid e_2^T \Delta p \mid + \mid e_3^T \Delta q \mid].$$

As above, using a variant of the Hölder inequality we obtain

$$\bar{\omega} \leq \omega[1-\alpha(1-\sigma)] + \alpha[\| e_1 \|_1 \| \alpha\Delta z \|_\infty + \| e_2 \|_1 \| \alpha\Delta p \|_\infty + \| e_3 \|_1 \| \alpha\Delta q \|_\infty].$$

But, from the specific choice of α we can write: $\| \alpha\Delta z \|_\infty \leq \| z \|_\infty \leq M$, $\| \alpha\Delta p \|_\infty \leq \| p \|_\infty \leq M$ and $\| \alpha\Delta q \|_\infty \leq \| q \|_\infty \leq M$ which was introduced into the last inequality, we can estimate the new complementarity as:

$$\bar{\omega} \leq \omega[1-\alpha(1-\sigma)] + M[\| e_1 \|_1 + \| e_2 \|_1 + \| e_3 \|_1]. \; \Box$$

As we know from the duality theory, three criteria must be met in order that a primal-dual solution is optimal: primal feasibility, dual feasibility and complementarity. The following theorems show the progress of the algorithm over several iterations with respect to the primal and dual feasability, as well as the complementarity.

Theorem 1 *Let $e_1^k(x,s)$, $e_2^k(x,w)$ and $e_3^k(x,v)$ denote the values of these quantities at the k-th iteration. Then:*

$$\begin{aligned} e_1^k(x,s) &\leq (1-\alpha)^k e_1^0(x,s), \\ e_2^k(x,w) &= (1-\alpha)^k e_2^0(x,w), \\ e_3^k(x,v) &= (1-\alpha)^k e_3^0(x,v). \end{aligned} \tag{7.9}$$

Proof From Proposition 9 we have $e_1^k(x,s) \leq (1-\alpha)e_1^{k-1}(x,s) \leq ... \leq (1-\alpha)^k e_1^0(x,s)$, which proves the first inequality of the proposition. Similarly, $e_2^k(x,w) = (1-\alpha)e_2^{k-1}(x,w) = ... \; (1-\alpha)^k e_2^0(x,w)$, and $e_3^k(x,v) = (1-\alpha)e_3^{k-1}(x,v) = ... = (1-\alpha)^k e_3^0(x,v)$. $\Box$

Theorem 2 *Let χ^k denote the value of χ at the k-th iteration. Then*

$$\chi^k \leq (1-\alpha)^k \chi^0. \tag{7.10}$$

Proof From Proposition 10 above we can write: $\chi^k \leq (1-\alpha)\chi^{k-1} \leq ... \leq (1-\alpha)^k \chi^0$, which proves the theorem. $\Box$

Theorem 3 *Let ω^k denote the value of ω at the k-th iteration. Assume that $\| z \|_\infty$, $\| p \|_\infty$, $\| q \|_\infty$ are bounded by a large real number $M < \infty$, and α defined by (6.17) is such that $\alpha \leq 1$. Then there exists a constant $\tilde{M} < \infty$ such that:*

$$\omega^k \leq (1-\bar{\alpha})^k \tilde{M} \tag{7.11}$$

where $\bar{\alpha} = \alpha(1-\sigma)$.

Proof From Proposition 12 and the previous estimations given by Theorem 1, we see that

$$\begin{aligned} \omega^k &\leq (1-\alpha(1-\sigma))\omega^{k-1} + M(1-\alpha)^{k-1}[\| e_1 \|_1 + \| e_2 \|_1 + \| e_3 \|_1] = \\ &(1-\bar{\alpha})\omega^{k-1} + \bar{M}(1-\alpha)^{k-1}, \end{aligned}$$

where $\bar{\alpha} = \alpha(1-\sigma)$ and $\bar{M} = M[\| e_1 \|_1 + \| e_2 \|_1 + \| e_3 \|_1]$. Since an analogous inequality relates ω^{k-1} to ω^{k-2}, we can substitute the corresponding inequality to get

$$\begin{aligned} \omega^k &\leq (1-\bar{\alpha})[(1-\bar{\alpha})\omega^{k-2} + \bar{M}(1-\alpha)^{k-2}] + \bar{M}(1-\alpha)^{k-1} = \\ &(1-\bar{\alpha})^2\omega^{k-2} + \bar{M}(1-\alpha)^{k-1}\left[\frac{1-\bar{\alpha}}{1-\alpha} + 1\right]. \end{aligned}$$

Continuing in this manner, we can write

$$\omega^k \leq (1-\bar{\alpha})^2[(1-\bar{\alpha})\omega^{k-3}+\bar{M}(1-\alpha)^{k-3}]+\bar{M}(1-\alpha)^{k-1}\left[\frac{1-\bar{\alpha}}{1-\alpha}+1\right]=$$
$$(1-\bar{\alpha})^3\omega^{k-3}+\bar{M}(1-\alpha)^{k-1}\left[\left(\frac{1-\bar{\alpha}}{1-\alpha}\right)^2+\frac{1-\bar{\alpha}}{1-\alpha}+1\right]\leq \ldots \leq$$
$$(1-\bar{\alpha})^k\omega^0+\bar{M}(1-\alpha)^{k-1}\left[\left(\frac{1-\bar{\alpha}}{1-\alpha}\right)^{k-1}+\ldots+\frac{1-\bar{\alpha}}{1-\alpha}+1\right].$$

But

$$(1-\alpha)^{k-1}\left[\left(\frac{1-\bar{\alpha}}{1-\alpha}\right)^{k-1}+\ldots+\frac{1-\bar{\alpha}}{1-\alpha}+1\right]=$$
$$\frac{\left(1-\bar{\alpha}\right)^k-(1-\alpha)^k}{\alpha-\bar{\alpha}}=\frac{\left(1-\bar{\alpha}\right)^k-(1-\alpha)^k}{\alpha\sigma}\leq\frac{\left(1-\bar{\alpha}\right)^k}{\alpha\sigma}.$$

Considering this last inequality , we see that

$$\omega^k \leq (1-\bar{\alpha})^k\omega^0+\bar{M}\frac{\left(1-\bar{\alpha}\right)^k}{\alpha\sigma}=(1-\bar{\alpha})^k\left[\omega^0+\frac{\bar{M}}{\alpha\sigma}\right].$$

Denoting $\left[\omega^0+\frac{\bar{M}}{\alpha\sigma}\right]=\widetilde{M}$, we see that $\omega^k \leq (1-\bar{\alpha})^k\,\widetilde{M}$. □

The above theorems are only a partial convergence result because all of them depend on the assumption that the step lengths remain bounded away from zero.

Remark Note that the primal and the dual infeasibilities go down by a factor of $(1-\alpha)$ at each iteration, whereas the complementarity goes down by the factor $(1-\bar{\alpha}) > (1-\alpha)$.

4. *The step length.* A crucial point of the algorithm is the step length determination. The following proposition shows that the sequence of $\{\bar{\alpha}_k\}$ generated by Step 5e of the PCNC algorithm is bounded away from zero.

Proposition 13 *Suppose that the functions of the problem $f(x)$, $g(x)$ and $h(x)$ are twice continuously differentiable and the derivative of $CO(t)$ given by (2.5) is Lipschitz continuous with a constant L. If $\{\sigma_k\}$ is bounded away from zero, then the sequence of $\{\bar{\alpha}_k\}$ generated by the PCNC algorithm is bounded away from zero.*

Proof Since $\bar{\alpha}_k=\min(\alpha_k^I,\alpha_k^{II})$ it suffices to show that the sequences $\{\alpha_k^I\}$ and $\{\alpha_k^{II}\}$ are bounded away from zero. Let us suppress the subscript k. Following El-Bakry, Tapia, Tsuchiya and Zhang [1996] let us define the vectors: $a(\alpha)=[s(\alpha)\ w(\alpha)\ v(\alpha)]^T \in R^{m+2n}$ and $b(\alpha)=[z(\alpha)\ p(\alpha)\ q(\alpha)]^T \in R^{m+2n}$. Hence the function $\Theta^I(\alpha)$ from (6.2) is written as:

$$\Theta^I(\alpha)=\frac{\min\{a_i(\alpha)b_i(\alpha)\}}{\frac{a^T(\alpha)b(\alpha)}{2n+m}}-\gamma\tau_1.$$

From the definition of α^I (Step 5c) we see that α is the largest number in $[0,\alpha_m]$ such that

$$a_i(\alpha)b_i(\alpha)-\gamma\tau_1 a^T(\alpha)b(\alpha)/(2n+m)\geq 0,$$

for every $i=1,\ldots,2n+m$ and $\alpha\in[0,\alpha^I]$.
Let $\eta_i=\mid \Delta a_i\Delta b_i-\gamma\tau_1\frac{\Delta a^T\Delta b}{2n+m}\mid$. From Proposition 8 it follows that Δt is bounded, then there is a positive constant M such that $\eta_i\leq M$. A straightforward computation shows that, for $\alpha\in[0,\alpha^I]$,

$$\begin{aligned}&a_i(\alpha)b_i(\alpha)-\gamma\tau_1\frac{a^T(\alpha)b(\alpha)}{2n+m}\\
=&\left[a_ib_i-\gamma\tau_1\frac{a^Tb}{2n+m}\right]+\alpha\left[a_i\Delta b_i+b_i\Delta a_i-\gamma\tau_1\frac{a^T\Delta b+b^T\Delta a}{2n+m}\right]+\alpha^2\left[\Delta a_i\Delta b_i-\gamma\tau_1\frac{\Delta a^T\Delta b}{2n+m}\right]\\
&\geq\alpha(1-\gamma\tau_1)\mu-\alpha^2\mid\Delta a_i\Delta b_i-\gamma\tau_1\frac{\Delta a^T\Delta b}{2n+m}\mid\\
&=\alpha(1-\gamma\tau_1)\mu-\eta_i\alpha^2\\
&\geq\alpha(1-\gamma\tau_1)\mu-M\alpha^2\geq 0.\end{aligned}$$

Hence

$$\alpha^I \geq (1-\gamma\tau_1)\mu/M.$$

But, as we know, $\mu = \sigma\frac{a^Tb}{2n+m}$ and for σ bounded away from zero it follows that μ is bounded below. Hence α^I is bounded away from zero.

Now let us show that $\{\alpha_k^{II}\}$ generated by Step 5 of the PCNC algorithm is bounded away from zero. According to the mean-value theorem for vector-valued function [Dennis and Schnabel, 1983, Chapter 4] we have:

$$\begin{aligned} CO(t+\alpha\Delta t) &= CO(t) + \alpha\left[\int_0^1 \nabla CO(t+\xi\alpha\Delta t)d\xi\right]\Delta t \\ &= CO(t) + \alpha\nabla CO(t)\Delta t + \alpha\left[\int_0^1 \left(\nabla CO(t+\xi\alpha\Delta t) - \nabla CO(t)\right)d\xi\right]\Delta t \\ &= (1-\alpha)CO(t) + \alpha\left[\int_0^1 \left(\nabla CO(t+\xi\alpha\Delta t) - \nabla CO(t)\right)d\xi\right]\Delta t, \end{aligned}$$

where the last equality is from (3.1). Now, having in view that the derivative of $CO(t)$ is Lipschitz continuous with the L constant, we obtain:

$$\| CO(t+\alpha\Delta t) \| \leq \| CO(t) \| | 1-\alpha | + L\alpha^2 \| \Delta t \|^2 .$$

Using this inequality we have:

$$\begin{aligned} \Theta^{II}(\alpha) &= a^T(\alpha)b(\alpha) - \gamma\tau_2 \| CO(t+\alpha\Delta t) \| \\ &= a^Tb + \alpha\left[a^T\Delta b + b^T\Delta a\right] + \alpha^2\Delta a^T\Delta b - \gamma\tau_2 \| CO(t+\alpha\Delta t) \| \\ &\geq | 1-\alpha | a^Tb + \alpha\sigma a^Tb + \alpha^2\Delta a^T\Delta b - \gamma\tau_2\left(\| CO(t) \| | 1-\alpha | + L\alpha^2 \| \Delta t \|^2\right) \\ &= | 1-\alpha | \left(a^Tb - \gamma\tau_2 \| CO(t) \|\right) + \alpha\sigma a^Tb + \alpha^2\left(\Delta a^T\Delta b - \gamma\tau_2 L \| \Delta t \|^2\right) \\ &\geq \alpha\left[\sigma a^Tb - \alpha \mid \Delta a^T\Delta b - \gamma\tau_2 L \| \Delta t \|^2 \mid\right]. \end{aligned}$$

From Proposition 8 there is a constant $N > 0$ such that

$$| \Delta a^T\Delta b - \gamma\tau_2 L \| \Delta t \|^2 | \leq N.$$

Hence

$$\Theta^{II}(\alpha) \geq \alpha\left[\sigma a^Tb - \alpha N\right].$$

From condition (6.15) it follows that

$$\alpha^{II} \geq \sigma a^Tb/N.$$

Since $\{\sigma_k\}$ is bounded away from zero, then the sequence $\{\alpha_k^{II}\}$ is bounded away from zero. □

Theorem 4 *Suppose that the functions of the problem $f(x)$, $g(x)$ and $h(x)$ are twice continuously differentiable and the derivative of $CO(t)$ given by (2.5) is Lipschitz continuous. Let $\{t_k\}$ be generated by the PCNC algorithm, where $\{\sigma_k\} \subset (0,1)$ is bounded away from zero and one. Then the sequence $\{F(t_k)\}$ converges to zero, and for any limit point $t^* = [x^*, y^*, z^*, p^*, q^*, s^*, w^*, v^*]^T$, x^* is a KKT point of the problem (1).*

Proof Firstly, note that the sequence $\{\| F(t_k) \|\}$ is monotonously decreasing, hence it is convergent. By contradiction, suppose that $\{\| F(t_k) \|\}$ is not convergent to zero. But, from Proposition 5 we have

$$\Phi_k(\alpha_k)/\Phi_k(0) \leq 1 - 2\alpha_k\beta(1-\sigma_k)$$

Then, from Proposition 13 it follows that the sequence $\{\Phi_k\}$ linearly converges to zero. This gives a contradiction.

On the other hand, from Proposition 2 we have:

$$\nabla\Phi(t_k)\Delta t_k = -2F(t_k)^TF(t_k) + 2\mu_kF(t_k)\hat{e}.$$

Since the sequence $\{\alpha_k\}$ is bounded away from zero, then the backtracking line search used at Step 5 of the PCNC algorithm produces:

$$\frac{\nabla\Phi(t_k)\Delta t_k}{\|\Delta t_k\|} = \frac{-2[F(t_k)^T F(t_k) - \mu_k F(t_k)\hat{e}]}{\|\Delta t_k\|} \to 0.$$

From Proposition 8, Δt_k is bounded away from zero, then

$$\Phi(t_k) - \mu_k[s_k^T z_k + w_k^T p_k + v_k^T q_k] \to 0.$$

However,

$$\Phi(t_k) - \mu_k[s_k^T z_k + w_k^T p_k + v_k^T q_k] \geq (1 - \sigma_k)\Phi(t_k).$$

Therefore, it must hold that $\Phi(t_k) \to 0$, because the sequence $\{\sigma_k\}$ is bounded away from one. Again this leads to a contradiction. Hence, the sequence $\{\| F(t_k) \|\}$ must converge to zero. Since the KKT conditions for the problem (1) are satisfied by t^*, it follows that x^* is a KKT point of the problem (1). $\square$

8. Numerical Examples

In this Section we will present some numerical results obtained with a crude experimental code which is still under development. Having in view that the conditions (6.17d) and (6.17f) are difficult to implement, for the step length determination, we shall consider a variant of the algorithm in which only the conditions (6.17a, b, c and e) are implemented. Firstly we present details of the algorithm running for a number of three problems. Afterwards, some characteristics of the optimization process with PCNC for some problems from the Hock and Schittkowski's and Schittkowski's sets of test problems, are presented.

Example 1. Let us consider the problem: [Hock and Schittkowski, 1981, problem 71, p.92]

$$\min x_1 x_4(x_1 + x_2 + x_3) + x_3$$

subject to:

$$\begin{aligned} x_1 x_2 x_3 x_4 - 25 &\geq 0, \\ x_1^2 + x_2^2 + x_3^2 + x_4^2 - 40 &= 0, \\ 1 \leq x_i \leq 5,\ i &= 1,2,3,4. \end{aligned}$$

The evolution of some elements of the PCNC algorithm is given in Tables 1 and 2.

Table 1. The Evolution of the Functions ($\varepsilon = 10^{-8}$):

$f(x_k)$ = *value of the objective*
$\Phi(t_k)$ = *value of the merit function*
$\| CO(t_k) \|_2$ = *norm of the optimality conditions*
$\| CT(.) \|_2$ = *norm of the transversality conditions*

k	$f(x_k)$	$\Phi(t_k)$	$\| CO(t_k) \|_2$	$\| CT(.) \|_2$
0	.160000E+02	.534076E+03	.465076E+03	.690000E+02
1	.169494E+02	.112666E+02	.897539E+01	.229126E+01
2	.171584E+02	.253076E+00	.897539E+01	.135262E+00
3	.169951E+02	.202962E-01	.141485E+01	.614777E-02
4	.170168E+02	.455528E-03	.188750E-03	.266778E-03
5	.170158E+02	.110764E-04	.526546E-06	.105499E-04
6	.170144E+02	.424528E-06	.561720E-09	.423877E-06
7	.170141E+02	.162452E-07	.651720E-09	.162445E-07
8	.170140E+02	.243738E-10	.199940E-14	.243718E-10

Table 2. The Evolution of the Parameters ($\varepsilon = 10^{-8}$):

γ_k = *value of parameter γ in function $\Theta^I(\alpha\ )$*

σ_k = *value of parameter σ for barrier parameter μ*

$\alpha_{\max}$ = *maximum value of the steplength*

α_k = *value of steplength*

μ_k = *value of the barrier parameter*

k	γ_k	σ_k	$\alpha_{\max}$	α_k	μ_k
0	0.750000	0.1	1.090858	1.039720	.233333E+00
1	0.625000	0.2	0.992283	0.926926	.906640E-01
2	0.562500	0.2	1.170237	1.0	.226439E-01
3	0.531250	0.2	1.210422	1.0	.488023E-02
4	0.515625	0.2	1.165723	1.0	.105805E-02
5	0.507812	0.2	1.220644	1.0	.216128E-03
6	0.503906	0.19530	1.238018	1.0	.423836E-04
7	0.501953	0.03823	1.039720	1.0	.162444E-05

Some remarks are in order:

1. The parameter γ_k is updated as: $\gamma_{k+1} = 0.5 + (\gamma_k - 0.5)/2$. Clearly, we can imagine some other updating formulas for $\gamma_k \in [0.5, \gamma_{k-1}]$.

2. The parameter σ_k is updated as:

$$\sigma_k = \begin{cases} \eta_1, & if \sigma_k \le \eta_2(s^Tz + w^Tp + v^Tq), \\ \eta_2(s^Tz + w^Tp + v^Tq), & if \sigma_k > \eta_2(s^Tz + w^Tp + v^Tq) \end{cases}$$

where $\eta_1 = 0.1$ and $\eta_2 = 100$.

3. Even though the condition (6.17d): $\Theta^{II}(\alpha_k) \geq 0$ has not been implemented, we see that in the last part of the optimization process: $\| CO(t_k) \|_2 < \| CT(.) \|_2$, this ensuring the convergence of the algorithm.

4. Note that the step length $\alpha_k \to 1$, exactly as in the "pure" Newton method.

5. It is interesting to see the evolution of the barrier parameter μ_k given by (5.4) or by (6.8). Table 3 illustrates this behaviour.

Table 3. The Evolution of the Barrier Parameter μ_k ($\varepsilon = 10^{-8}$)

k	μ_k given by (6.8)	μ_k given by (5.4)
0	.233333E+00	.254322E+02
1	.906640E-01	.276151E+01
2	.226439E-01	.248363E+00
3	.488023E-02	.924194E-01
4	.105805E-02	.956741E-02
5	.216128E-03	.113888E-02
6	.423836E-04	.217363E-03
7	.162444E-05	.424866E-04

6. The solution of this problem is $x = [1\ 5\ 5\ 1]^T$. The value of the objective function is 17.0140173. The value of the equality constraints is .7638083E-08, and of the inequality constraints is .2935759E-05.

7. The SPENBAR package [Andrei, 1996a-d] gives a solution for this problem, involving 8 major iterations, 143 minor iterations (truncated Newton iterations) and 591 evaluations of the functions of the problem.

Example 2. Let us consider the problem [Hock and Schittkowski, 1981, problem 43, p.66 (Rosen -Suzuki)]:

$$\min x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$$

subject to:

$$\begin{aligned} 8 - x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 \geq 0, \\ 10 - x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 \geq 0, \\ 5 - 2x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 \geq 0, \\ -2 \leq x_i \leq 5,\ i = 1, 2, 3, 4. \end{aligned}$$

The PCNC algorithm gives the following solution in Table 4:

Table 4. Solution of the Problem ($\varepsilon = 10^{-8}$)

#	l	x^*	u
1	-2.0	.992394E-12	5.0
2	-2.0	1.0	5.0
3	-2.0	2.0	5.0
4	-2.0	-1.0	5.0

The evolution of some elements of the PCNC algorithm is given in Tables 5 and 6.

Table 5. The Evolution of the Functions ($\varepsilon = 10^{-8}$):

$f(x_k)$ = *value of the objective*
$\Phi(t_k)$ = *value of the merit function*
$\| CO(t_k) \|_2$ = *norm of the optimality conditions*
$\| CT(.) \|_2$ = *norm of the transversality conditions*

k	$f(x_k)$	$\Phi(t_k)$	$\| CO(t_k) \|_2$	$\| CT(.) \|_2$
0	.000000E+00	.737000E+03	.536000E+03	.201000E+03
1	-.977641E+01	.549977E+03	.401579E+03	.148398E+03
2	-.371578E+02	.100816E+03	.528640E+02	.479521E+02
3	-.445550E+02	.783561E+01	.259689E+01	.523872E+01
4	-.437880E+02	.341158E+00	.413957E-01	.299762E+00
5	-.439719E+02	.322406E-01	.100845E-02	.312322E-01
6	-.439805E+02	.136610E-02	.150560E-03	.121554E-02
7	-.439962E+02	.480056E-04	.819310E-07	.479236E-04
8	-.439991E+02	.191398E-05	.441733E-10	.191394E-05
9	-.439998E+02	.768899E-07	.259717E-13	.768898E-07
10	-.439999E+02	.656910E-09	.612515E-16	.656910E-09
11	-.440000E+02	.531802E-13	.681308E-20	.531802E-13

Table 6. The Evolution of the Parameters ($\varepsilon = 10^{-8}$):

γ_k = *value of parameter γ in function $\Theta^I(\alpha\ )$*
σ_k = *value of parameter σ for barrier parameter μ*
$\alpha_{\max}$ = *maximum value of the steplength*
α_k = *value of steplength*
μ_k = *value of the barrier parameter*

k	γ_k	σ_k	α_{max}	α_k	μ_k
0	.750000	.3	0.173623	0.173620	.106363E+01
1	.625000	.2	0.610374	0.610371	.615770E+00
2	.562500	.2	0.776175	0.776172	.350629E+00
3	.531250	.2	1.099367	1.000000	.116627E+00
4	.515625	.2	0.888802	0.888801	.284905E-01
5	.507812	.2	1.170841	1.0	.888200E-02
6	.503906	.2	1.164319	1.0	.203728E-02
7	.501953	.2	1.193870	1.0	.415577E-03
8	.500976	.2	1.243771	1.0	.834214E-04
9	.500488	.91966E-01	1.101126	1.0	.768897E-05
10	.500244	.85005E-02	1.008572	1.0	.656909E-07
11	.500122	.76484E-04	1.000076	1.0	.531801E-11

The SPENBAR package gives the same solution involving 5 major iterations, 85 minor iterations and 347 functions evaluations.

Example 3. Let us consider the problem [Hock and Schittkowski, 1981, problem 113, p.122 (Wong No. 2)]:

$$\min x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$$

subject to:

$$\begin{gathered} 105 - 4x_1 - 5x_2 + 3x_7 - 9x_8 \geq 0, \\ -10x_1 + 8x_2 + 17x_7 - 2x_8 \geq 0, \\ 8x_1 - 2x_2 - 5x_9 + 2x_{10} + 12 \geq 0, \\ -3(x_1 - 2)^2 - 4(x_2 - 3)^2 - 2x_3^2 + 7x_4 + 120 \geq 0, \\ -5x_1^2 - 8x_2 - (x_3 - 6)^2 + 2x_4 + 40 \geq 0, \\ -3x_5^2 - 0.5(x_1 - 8)^2 - 2(x_2 - 4)^2 + x_6 + 30 \geq 0, \\ -x_1^2 - 2(x_2 - 2)^2 + 2x_1x_2 - 14x_5 + 6x_6 \geq 0, \\ 3x_1 - 6x_2 - 12(x_9 - 8)^2 + 7x_{10} \geq 0, \\ 0 \leq x_i \leq 10,\ i = 1, ..., 10. \end{gathered}$$

The PCNC algorithm gives the following solution in Table 7:

Table 7. Solution of the Problem ($\varepsilon = 10^{-8}$)

#	l	x^0	x^*	u
1	0.0	2.0	.2171996E+01	10.0
2	0.0	3.0	.2363683E+01	10.0
3	0.0	5.0	.8773926E+01	10.0
4	0.0	5.0	.5095984E+01	10.0
5	0.0	1.0	.9906548E+01	10.0
6	0.0	2.0	.1430574E+01	10.0
7	0.0	7.0	.1321644E+01	10.0
8	0.0	3.0	.9828726E+01	10.0
9	0.0	6.0	.8280092E+01	10.0
10	0.0	10.0	.8375927E+01	10.0

Table 8. The Value of the Constraints ($\varepsilon = 10^{-8}$)

i	$h_i(x^*)$
1	.3936866E-09
2	.1424133E-08
3	.4911662E-09
4	.3292851E-07
5	.2165631E-08
6	.6148503E+01
7	.2354256E-08
8	.5002396E+02

The value of the objective function is: 24.30621.
The evolution of some elements of the PCNC algorithm is given in Tables 9 and 10.

Table 9. The Evolution of the Functions ($\varepsilon = 10^{-8}$):

$$
\begin{aligned}
f(x_k) &= \textit{value of the objective} \\
\Phi(t_k) &= \textit{value of the merit function} \\
\| CO(t_k) \|_2 &= \textit{norm of the optimality conditions} \\
\| CT(.) \|_2 &= \textit{norm of the transversality conditions}
\end{aligned}
$$

k	$f(x_k)$	$\Phi(t_k)$	$\| CO(t_k) \|_2$	$\| CT(.) \|_2$
0	.753000E+03	.451040E+05	.442410E+05	.863000E+03
1	.576916E+03	.318551E+05	.312051E+05	.649950E+03
2	.221882E+03	.760135E+04	.740377E+04	.197577E+03
3	.790325E+02	.915963E+03	.878209E+03	.377534E+02
4	.447106E+02	.342317E+03	.331333E+03	.109840E+02
5	.265738E+02	.270729E+02	.261570E+02	.915983E+00
6	.245296E+02	.592638E-01	.162438E-01	.430200E-01
7	.243544E+02	.186719E-02	.340773E-04	.183312E-02
8	.243160E+02	.742385E-04	.109610E-06	.741289E-04
9	.243081E+02	.298381E-05	.102978E-09	.298371E-05
10	.243066E+02	.119888E-06	.106694E-12	.119888E-06
11	.243062E+02	.404292E-08	.171373E-15	.404292E-08
12	.243062E+02	.470981E-11	.376155E-18	.470981E-11

Table 10. The Evolution of the Parameters ($\varepsilon = 10^{-8}$):

$$
\begin{aligned}
\gamma_k &= \textit{value of parameter } \gamma \textit{ in function } \Theta^I(\alpha\ ) \\
\sigma_k &= \textit{value of parameter } \sigma \textit{ for barrier parameter } \mu \\
\alpha_{\max} &= \textit{maximum value of the steplength} \\
\alpha_k &= \textit{value of steplength} \\
\mu_k &= \textit{value of the barrier parameter}
\end{aligned}
$$

k	γ_k	σ_k	$\alpha_{\max}$	α_k	μ_k
0	.750000	.3	0.158304	0.158304	.140357E+01
1	.625000	.2	0.502918	0.502918	.809315E+00
2	.562500	.2	0.657969	0.657969	.435822E+00
3	.531250	.2	0.535589	0.535589	.200213E+00
4	.515625	.2	0.890842	0.890842	.113156E+00
5	.503906	.2	1.127216	1.0	.341684E-01
6	.503906	.2	1.209376	1.0	.764177E-02
7	.501953	.2	1.170946	1.0	.160451E-02
8	.500976	.2	1.222116	1.0	.325237E-03
9	.500488	.2	1.244544	1.0	.652865E-04
10	.500244	.18321	1.223653	1.0	.119888E-04
11	.500122	.33645	1.034818	1.0	.404292E-06
12	.500061	.11483	1.001150	1.0	.470981E-09

In this case, the SPENBAR package requires 4 major iterations, 149 minor (truncated Newton) iterations and 970 evaluations of functions.

In the following we shall consider the running of the PCNC package on a number of problems from the Hock - Schittkowski [1981] set of problems, as well as from Schittkowski's [1987] collection of problems. Table 11 shows the characteristics of the optimization process with PCNC referring to different initial points, number of iterations, number of functions evaluations, the value of the norm of the optimality conditions, the value of the barrier parameter and of the step length, etc.

Table 11. Characteristics of the Optimization Process with PCNC ($\varepsilon = 10^{-8}$):

$$
\begin{aligned}
IP &= \textit{Initial Point } (S = \textit{Standard Point}) \\
nit &= \#\ \textit{of iterations} \\
nf &= \#\ \textit{of functions evaluations} \\
\| CO(t_k) \|_2 &= \textit{Norm of optimality conditions} \\
\| CT(.) \|_2 &= \textit{Norm of centrality conditions} \\
f &= \textit{Value of the objective function} \\
\xi &= \textit{Distance from centrality} \\
\mu &= \textit{Value of the barrier parameter} \\
\alpha &= \textit{Value of the steplength} \\
\alpha_{\max} &= \textit{Maximum value of the steplength}
\end{aligned}
$$

$Name$	IP	nit	nf	$\| CO(t_k) \|_2$	$\| CT(.) \|_2$
HS32	S	6	8	.281190E-09	.890623E-5
	I1	6	8	.827097E-08	.906026E-04
	I2	35	37	.238509E-09	.706660E-05
	I3	36	38	.542306E-08	.512380E-04
HS33	S	61	63	.723080E-08	.346289E-11
	I1	57	59	.238560E-08	.186609E-12
HS37	I1	14	16	.206426E-27	.228163E-20
HS42	S	10	12	.476987E-26	.106282E-16
	I1	9	11	.263809E-24	.572985E-15
	I2	10	12	.232637E-28	.325052E-19
	I3	9	11	.137112E-24	.297692E-15
HS43	S	13	15	.277706E-09	.302020E-09
	I1	12	14	.134241E-09	.139405E-09
	I2	12	14	.246198E-09	.256468E-09
	I3	13	15	.121085E-09	.127335E-09
HS44	I1	12	14	.789613E-29	.906522E-19
	I2	12	14	.342044E-28	.539278E-18
HS47	I1	55	57	.867998E-08	.883014E-41
	I2	54	56	.307564E-04	.267626E-18
HS48	S	4	6	.981479E-13	.380432E-01
	I1	4	6	.935982E-13	.376255E-01
	I2	4	6	.328648E-10	.460264E-01
HS49	S	15	17	.914903E-08	.240757E-11
	I1	26	28	.277362E-08	.171366E-19
HS50	S	10	12	.207630E-09	.231393E-06
	I1	9	11	.180172E-11	.123154E-05
	I2	14	16	.275918E-13	.567711E-20
	I3	6	8	.114298E-09	.968789E+00
	I4	6	8	.644158E-08	.250051E+00
HS51	S	4	6	.385563E-12	.115810E-01
	I1	4	6	.551270E-11	.283667E-01
	I2	4	6	.175057E-11	.196812E-01
HS52	S	2	4	.347500E-09	.161794E+03
	I1	3	5	.195840E-08	.423945E+02
	I2	3	5	.223451E-08	.426037E+02
	I3	8	10	.116096E-13	.311722E+01
HS53	S	3	5	.155242E-10	.395810E+00
	I1	4	6	.135780E-08	.223385E+01
	I2	4	6	.247691E-08	.257194E+01
	I3	3	5	.134293E-11	.729796E-01
HS61	I1	9	11	.255905E-10	.485008E-08
HS63	S	6	8	.307336E-10	.155108E-04
	I1	6	8	.707607E-08	.124090E-04
	I2	9	11	.511842E-11	.744801E-05
HS65	S	7	9	.276898E-09	.102787E-04
	I1	7	9	.673553E-11	.123496E-05
	I2	5	7	.648425E-08	.283699E-04
HS71	S	6	8	.651720E-09	.423877E-06
	I1	7	9	.435015E-08	.884619E-06
HS73	S	8	10	.176433E-10	.173662E-05
	I1	10	12	.513410E-09	.120897E-08
	I2	9	11	.339001E-08	.442336E-06
	I3	10	12	.712451E-09	.290912E-06
HS76	S	10	12	.140067E-29	.343496E-16
	I1	10	12	.578975E-29	.752265E-15
	I2	11	13	.108198E-29	.104317E-20
	I3	12	14	.188957E-29	.814056E-17
HS113	S	13	15	.366188E-25	.487388E-18
	I1	17	19	.167600E-22	.204212E-15
	I2	12	14	.176304E-23	.224720E-16
	I3	18	20	.248136E-25	.269701E-18
S217	S	13	15	.749654E-24	.615541E-19
	I1	9	11	.112356E-23	.923683E-19
S218	S	14	16	.268757E-10	.314477E-07
S219	I1	14	16	.388070E-10	.163815E-09

Table 11. Characteristics of the Optimization Process with PCNC ($\varepsilon = 10^{-8}$) - continued

Name	f	ξ	μ	α	$\alpha_{\max}$
HS32	1.0	.673457	.747336E-03	0.999500	1.287417
	1.0	.631756	.281243E-02	0.999500	1.219009
	1.0	..622537	.626097E-02	0.999500	1.266965
	1.0	.704269	.184727E-02	0.999500	1.229708
HS33	1.798992	.00022	.374461E-08	0.717951	0.718310
	1.798992	0.00047	.141810E-08	0.893644	0.894091
HS37	-3456.0	0.999999	.585360E-12	0.999397	1.000
HS42	13.85786	0.99995	.688951E-09	0.9995	1.00074
	13.85786	0.999949	.698862E-08	0.9995	1.00237
	13.85786	0.99995	.943834E-11	0.9995	1.00008
	13.85786	0.999949	.486897E-08	0.9995	1.00197
HS43	-44.0	.282086	.668179E-08	0.135259	1.00829
	-44.0	0.166757	.879352E-08	0.630348	1.014314
	-44.0	0.177464	.531238E-08	0.212828	1.012284
	-44.0	0.203728	.212161E-08	0.086971	1.006461
HS44	-15.0	.999473	.202883E-10	0.999500	1.000169
	-15.0	.999473	.781634E-10	0.999500	1.000331
HS47	0.0	.04927	.546914E-17	0.938531	0.999661
	.133442E-03	0.05	.221299E-16	.312620E-03	0.991663
HS48	0.0	.999169	.615545E-04	0.999500	1.223207
	0.0	.999211	.612159E-04	0.999500	1.222694
	0.0	.990773	.213457E-03	0.999500	1.140494
HS49	0.0	.0495064	.180460E-10	0.093525	0.997787
	0.0	.150	.222354E-16	0.765397	0.997641
HS50	0.0	.999374	.151811E-03	0.999500	1.245079
	0.0	.999917	.350233E-03	0.999500	1.247689
	0.0	.883577	.190954E-11	0.999500	1.000040
	.487294E-07	.999076	.310624E+00	0.999500	1.210478
	.205908E-06	.997404	.498941E+00	0.999500	1.183567
HS51	.331885E-05	.948848	.334951E-01	0.999500	1.149867
	.113598E-04	.934210	.522127E-01	0.999500	1.162979
	.348339E-06	.959153	.441287E-01	0.999500	1.150994
HS52	5.326641	.999659	.401433E+01	0.999500	1.219962
	5.326636	.999791	.205487E+01	0.999500	1.229861
	5.326659	.999790	.205994E+01	0.999500	1.229707
	5.326648	.999996	.557206E+00	0.999500	1.249595
HS53	4.093025	.996253	.198503E+00	0.999500	1.160760
	4.093031	.996831	.471426E+00	0.999500	1.164569
	4.093054	.992703	.505691E+00	0.999500	1.169289
	4.093024	.998795	.852519E-01	0.9995000	1.232501
HS61	-143.64614	.999699	.283372E-04	0.999500	1.144733
HS63	961.7152	.992672	.160669E-02	0.999500	1.231509
	961.7151	.934392	.145152E-02	0.999500	1.174307
	961.7152	.999600	.111198E-02	0.999500	1.239978
HS65	.9547013	.966343	.120727E-02	0.999500	1.215245
	.9539442	.988519	.418477E-03	0.999500	1.208355
	.9554332	.946089	.198347E-02	0.999500	1.187995
HS71	17.01444	.984698	.216128E-03	0.999500	1.220644
	17.01462	.975000	.311775E-03	0.999500	1.214727
HS73	29.89562	.964530	.416176E-03	0.999500	1.097279
	29.89441	.995412	.109476E-04	0.999500	1.115710
	29.89500	.970094	.209943E-03	0.999500	1.143323
	29.89489	.994730	.170205E-03	0.999500	1.136257
HS76	-4.681818	.998205	.123711E-08	0.999500	1.001168
	-4.681818	.998201	.700899E-08	0.999500	1.002785
	-4.681818	.998200	.384940E-12	0.999500	1.000021
	-4.681818	.998207	.517577E-09	0.999500	1.000755
HS113	24.30621	.999871	.592351E-10	0.999500	1.000407
	24.30621	.999870	.225305E-08	0.999500	1.002518
	24.30621	.999870	.654417E-09	0.999500	1.001356
	24.30621	.999871	.390857	0.999500	1.000331
S217	-0.8	.997026	.174759E-10	0.999500	1.000094
	-0.8	.997027	.247263E-10	0.999500	1.000111
S218	.774652E-07	.0220820	.105397E-08	0.225725	1.031481
S219	-0.999999	.100000	.570597E-08	0.089401	1.021684

Table 12 shows the comparative results of the PCNC and the SPENBAR results of the set of problems considered in this experiment. In PCNC code the Newton method is used as the basic iterative procedure. In SPENBAR we used the truncated Newton method [Nash, 1985], [Nash, Polyak and Sofer, 1994]. In the modified penalty-barrier code SPENBAR, the number of major iterations *(o-it)* is the number of times μ (the barrier parameter) was decreased and the Lagrange multipliers adjusted. The number of minor iterations *(in-it)* is the total number of truncated Newton steps. For the PCNC, the number of iterations is the total number of Newton steps. In both codes , *nf* is the total number the functions of the problem has been evaluated.

Table 12: Comparison Between SPENBAR and PCNC

Problem	n	m	me	SPENBAR			PCNC	
				o-it	in-it	nf	it	nf
HS32	3	1	1	10	136	348	6	8
HS33	3	2	0	10	66	187	61	63
HS37	3	2	0	4	26	76	14	16
HS42	4	0	2	5	23	84	10	12
HS43	4	3	0	5	85	347	13	15
HS44	4	6	0	9	107	498	12	14
HS47	5	0	3	8	28	278	55	57
HS48	5	0	2	4	50	179	4	6
HS49	5	0	2	3	126	459	15	17
HS50	5	0	3	4	106	388	10	12
HS51	5	0	3	5	91	324	4	6
HS52	5	0	3	6	81	280	2	4
HS53	5	0	3	6	74	284	3	5
HS61	3	0	2	4	37	96	9	11
HS63	3	0	2	4	51	153	6	8
HS65	3	1	0	5	203	659	7	9
HS71	4	1	1	8	143	591	6	8
HS73	4	2	1	7	285	1128	8	10
HS76	4	3	0	9	96	364	10	12
HS113	10	8	0	4	149	970	13	15
S217	2	1	1	5	33	83	13	15
S218	2	1	0	9	56	234	14	16
S219	4	0	2	6	105	401	14	16
TOTAL				140	2152	8411	309	355

Computational experience with this approach is very limited. Using the same primal-dual interior-point approach, El-Bakry, Tapia, Tsuchiya and Zhang [1996] report some preliminary computational results on a limited number of small scale test problems from the Hock and Schittkowski collection. Lasdon, Plummer and Yu [1995] report results on a somewhat larger test set, of a variant of algorithm using a trust region approach. Vanderbei and Shanno [1997] present results on the Hock - Schittkowski problems as well as on a number of 8 large-scale nonlinear problems, in comparison with MINOS and LANCELOT. Gay, Overton and Wright [1997] also report results on the Hock - Schittkowski problems and three problems from the CUTE [1995] collection. We have considered here some numerical experience with a variant of the algorithm, which is a mere simplification, i.e. we did not enforce conditions (6.17d and 6.17f) for step length determination in order to avoid the complications regarding the nonlinear function $\Theta^{\mu}(\alpha)$. A robust implementation must consider these conditions in the line search procedure from step 5 of the algorithm. The algorithm contains a number of parameters, and its performance is largely dependent on their choice. To choose proper values of these parameters requires very intensive numerical tests.

9. Conclusion

In this paper we have presented an interior point algorithm for solving general nonlinear programming problems. It is based on the perturbed Karush-Kuhn-Tucker conditions associated with the logarithmic barrier function formulation of the problem.

The algorithm has two main components: determination of the descent direction and the computation of the step length. The search direction is computed as a solution for a reduced perturbed Newton system. Vanderbei and Shanno [1997] show that for nonconvex nonlinear problems this system must be modified by adding a diagonal matrix. We did not consider this perturbation in our implementation. The critical feature of the algorithm is the choice of the step lengths α_k at each iteration, in order to find the local minimizer of (1). For this, a number of criteria must be implemented (see (6.17)), the main one is that referring to the speed of reduction to zero of the pure optimality conditions versus the transversality conditions. The convergence is ensured when both these conditions converge to zero, but in such a proportion that the transversality conditions surmount the pure optimality conditions. In our experimental code we did not implement this condition.

Mainly, the algorithm does not mimic the primal-dual interior-point methods for linear programming. Rather some special elements have been introduced as: formulation of the optimality conditions, definition of the perturbed reduced Newton system, definition of a merit function, choice of the barrier parameter. Some other points remain to be clarified: detection and treatment of the indefiniteness in the Newton system, selection of proper values of parameters β, γ_k, σ_k and ρ, treatment of an infeasible starting point, practical implementation of the conditions (6.17) for the step length determination. All of these issues deserve further research and experimentation.

The results of this preliminary computational experimentation indicate that the primal-dual approach is more efficient than the penalty-barrier approach. However, the primal-dual algorithm requires second partial derivatives of the functions of the problem. Thus, for those problems for which the second derivative information is difficult to obtain, the penalty-barrier method should be preferred. One advantage of the primal-dual approach is that the Lagrange multipliers are computed directly by the Newton's method rather than by using the first order estimates. This is very appealing, since improving both the accuracy and the rate of convergence of the algorithm.

In summary, the reconsideration of the barrier methods for nonlinear programming is one of the most active areas of research. A lot remains to be done from both theoretical and computational viewpoints, in order to build up a viable algorithm and to achieve an efficient implementation, able to enter the numerical comparisons with the state-of-the-art packages such as: MINOS, LANCELOT, SNOPT, NLPQL, SPENBAR, CONOPT, etc.

Acknowledgements

I gratefully acknowledge the Alexander von Humboldt Foundation for their constantly and generously given moral, financial and material support for the two years which the author spent at Duisburg and Bayreuth Universities, Germany.

Special thanks are due to Professor Klaus Schittkowski (Bayreuth University) for his stimulating discussions and availability during all my stay in Bayreuth, Professor Michael Saunders (Stanford University) for providing the MINOS code, as well as some other practical hints at its usage, and Dr. Marc Breitfeld and Professor David Shanno (Rutgers University) for the PENBAR package and some papers on penalty-barrier algorithms for nonlinear programming.

REFERENCES

ADLER, I., KARMARKAR, N., RESENDE , M.G.C. and VEIGA, G., **Data Structures and Programming Techniques for the Implementation of Karmarkar's Algorithm**, ORSA JOURNAL ON COMPUTING, Vol.1, 1989a, pp.84-106.

ADLER, I., KARMARKAR, N., RESENDE, M.G.C. and VEIGA, G., **An Implementation of Karmarkar's Algorithm for Linear Programming**, MATHEMATICAL PROGRAMMING, Vol.44, 1989b, pp.297-335.

ANDERSEN, E.D.and ANDRESEN, K.D., **The APOS Linear Programming Solver: An Implementation of the Homogeneous Algorithm**, Technical Report, April 28, 1997, Department of Management, Odense University DK-5230 Odense, and CORE, Universite Catholique de Louvain, Belgium.

ANDERSEN, E.D., GONDZIO, J., MESZAROS, C. and XU, X., **Implementation of Interior-point Methods for Large Scale Linear Programs**, in T. Terlaky (Ed.) Interior Point Methods of Mathematical Programming, KLUWER ACADEMIC PUBLISHERS, Dordrecht, 1996, pp.189-252.

ANDREI, N., **Computational Experience with A Modified Penalty-barrier Method for**

Large-scale Nonlinear Constrained Optimization, ICI Working Paper No. AMOL-96-1, February 6, 1996a.

ANDREI, N., **Computational Experience with A Modified Penalty-barrier Method for Large-scale Nonlinear Equality and Inequality Constrained Optimization**, ICI Technical Paper, No. AMOL-96-2, February 12, 1996b.

ANDREI, N., **Computational Experience with SPENBAR - A Sparse Variant of A Modified Penalty-barrier Method for Large-scale Nonlinear Equality and Inequality Constrained Optimization**, ICI Working Paper, No. AMOL-96-3, March 10, 1996c.

ANDREI, N., **Numerical Examples with SPENBAR for Large-scale Nonlinear, Equality and Inequality, Constrained Optimization with Zero Columns in Jacobian Matrices**, ICI Technical Paper No. AMOL-96-5, March 29, 1996d.

ANDREI, N., **PCIP - Predictor Corrector Interior Point Algorithm for Large-scale Linear Programming**, Technical Paper No. AMOL-97-1, Research Institute for Informatics, Bucharest, March 1997a.

ANDREI, N., **PCIPLC - Predictor Corrector Interior Point Algorithm for Large-scale Linear Constrained Optimization Problems**, Technical Paper No. AMOL-97-2, Research Institute for Informatics, Bucharest, May 1997b.

ANDREI, N., **Penalty-barrier Algorithms for Nonlinear Optimization. Preliminary Computational Results**, STUDIES IN INFORMATICS AND CONTROL, Vol.7, No.1, March 1998a, pp. 15-36.

ANDREI, N., **Predictor-Corrector Interior-point Method for Linear Constrained Optimization**, STUDIES IN INFORMATICS AND CONTROL, Vol.7, No. 2, June 1998b, pp.155-177.

ARGAEZ, M. and TAPIA, R.A., **On the Global Convergence of A Modified Augmented Lagrangian Linesearch Interior Point Newton Method for Nonlinear Programming**, Report 95-38 (modified February 1997), Department of Computational and Applied Mathematics, Rice University, Houston, TXS, 1997.

BONGARTZ, I., CONN, A.R., GOULD, N.I.M.and TOINT, PH.L., **CUTE: Constrained and Unconstrained Testing Environment**, ACM TRANS. MATH. SOFTWARE, Vol.21, 1995, pp.123-160.

BROOKE, A., KENDRICK, D.and MEERAUS, A., **GAMS - A User's Guide**, Release 2.25, THE SCIENTIFIC PRESS, South San Francisco, CA, 1992.

BYRD, R.H., GILBERT, J.C. and NOCEDAL, J., **A Trust Region Method Based On Interior Point Techniques for Nonlinear Programming**, Technical Report OTC 96/02, Argonne National Laboratory, Argonne, ILL, 1996.

BYRD, R.H., HRIBAR, M.E. and NOCEDAL, J., **An Interior Point Algorithm for Large Scale Nonlinear Programming**, Technical Report OTC 97/05, Argonne National Laboratory, Argonne, ILL, 1997.

CARPENTER, T.J., LUSTIG, I.J., MULVEY, J.M. and SHANNO, D.F., **Higher-order Predictor-corrector Interior Point Methods with Application to Quadratic Objectives**, SIAM JOURNAL ON OPTIMIZATION, Vol.3, No.4, November 1993, pp.696-725.

CHOI, I.C., MONMA, C.L. and SHANNO, D.F., **Further Development of A Primal Dual Interior Point Method**, ORSA JOURNAL ON COMPUTING, Vol.2, 1990, pp.304-311.

CONN, A.R., GOULD, N.I.M. and TOINT, PH. L., **LANCELOT - A Fortran Package for Large-scale Nonlinear Optimization (Release A)**, SPRINGER-VERLAG, Berlin, 1992.

CONN, A.R., GOULD, N.I.M. and TOINT, PH. L., **A Primal-dual Algorithm for Minimizing A Nonconvex Function Subject to Bound and Linear Equality Constraints**, Report RC 20639, IBM T.J. Watson Research Center, Yorktown Heights, New York, 1996.

DENNIS, J.E., Jr.and SCHNABEL, R.B., **Numerical Methods for Unconstrained Optimization**, PRENTICE-HALL, Englewood Cliffs, NJ, 1983.

DRUD, A., **CONOPT - A System for Large Scale Nonlinear Optimization, Reference Manual for CONOPT Subroutine Library**, ARKI Consulting and Development A/S, Bagsvared, Denmark, 1996.

EL-BAKRY, A.S., TAPIA, R.A., TSUCHIYA, T. and ZHANG, Y., **On the Formulation and Theory of the Newton Interior Point Method for Nonlinear Programming**, JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS, Vol.89, No.3, 1996, pp.507-541.

FIACCO, A.V. and MCCORMICK, G.P., **Nonlinear Programming. Sequential Unconstrained Minimization Techniques**, JOHN WILEY & SONS, New York, 1968.

FORSGREN, A.and GILL, P.E., **Primal-dual Interior Methods for Nonconvex Nonlinear**

Programming, Report NA 96-3, Department of Mathematics, University of California, San Diego, 1996.

FOURER, R., GAY, D.M.and KERNIGHAN, B.W., **AMPL: A Modeling Language for Mathematical Programming**, THE SCIENTIFIC PRESS, 1993.

GAY, D.M., OVERTON, M.L. and WRIGHT, M.H., **A Primal-Dual Interior -Method for Nonconvex Nonlinear Programming**, Technical Report 97-4-08, Bell Laboratories, Murray Hill, NJ, July 29, 1997.

GILL, PH. E., MURRAY, W., PONCELEON, D.B.and SAUNDERS, M.A., **Primal-dual Methods for Linear Programming**, MATHEMATICAL PROGRAMMING, Vol.70, 1995, pp.251-277.

GILL, PH. E., MURRAY, W. and SAUNDERS, M.A., **User Guide for SQOPT 5.3: A FORTRAN Package for Large-Scale Linear and Quadratic Programming**, Technical Report, Systems Optimization Laboratory, Department of EESOR, Stanford University, Stanford, CA, 94305-4023, October 1997.

GILL, PH. E., MURRAY, W. and SAUNDERS, M.A., **User's Guide for SNOPT 5.3: A FORTRAN Package for Large-Scale Nonlinear Programming**, Technical Report, Systems Optimization Laboratory, Department of EESOR, Stanford University, Stanford, CA, 94305-4023, March 4, 1998.

GOLDFARB, D., LIU, S. and WANG, S., **A Logarithmic Barrier Function Algorithm for Quadratically Constrained Convex Quadratic Programming**, SIAM JOURNAL ON OPTIMIZATION, Vol.1, No.2, 1991, pp.252-267.

GONDZIO, J., **Multiple Centrality Corrections in A Primal-dual Method for Linear Programming**, COMPUTATIONAL OPTIMIZATION AND APPLICATIONS, Vol.6, No.2, September 1996, pp.137-156

HOCK, W. and SCHITTKOWSKI, K., **Test Examples for Nonlinear Programming Codes**, SPRINGER- VERLAG, Berlin, Heidelberg, New York, 1981.

KORTANEK, K.O., POTRA, F.A. and YE, Y., **On Some Efficient Interior Point Methods for Nonlinear Convex Programming**, LINEAR ALGEBRA AND ITS APPLICATIONS, Vol.152, 1991, pp.169-189.

LASDON, L.S., PLUMMER, J. and YU, G., **Primal-dual and Primal Interior Point Algorithms for General Nonlinear Programs**, ORSA JOURNAL ON COMPUTING, Vol.7, 1995, pp.321-332.

LUSTIG, I.J., MARSTEN, R.E. and SHANNO, D.F., **The Primal-dual Interior Point Method on the Cray Supercomputer**, in T. F. Coleman and Y. Li (Eds.) Large-Scale Numerical Optimization, SIAM, Philadelphia, 1990, pp.70-80.

LUSTIG, I.J., MARSTEN, R.E. and SHANNO, D.F., **Computational Experience with A Primal-dual Interior Point Method for Linear Programming**, LINEAR ALGEBRA AND ITS APPLICATIONS, Vol.152, 1991, pp.191-222.

LUSTIG, I.J., MARSTEN, R.E.and SHANNO, D.F., **On Implementing Mehrotra's Predictor-corrector Interior-point Method for Linear Programming**, SIAM JOURNAL ON OPTIMIZATION, Vol.2, No.3, August 1992, pp.435-449.

LUSTIG, I.J., MARSTEN, R.E. and SHANNO, D.F., **Computational Experience with A Globally Convergent Primal-dual Predictor-corrector Algorithm for Linear Programming**, MATHEMATICAL PROGRAMMING, Vol.66, 1994, pp.123-135.

MESZAROS, C., **Fast Cholesky Factorization for Interior Point Methods of Linear Programming**, Technical Report, Computer and Automation Research Institute, Budapest, 1994.

MITTELMANN, H., **Benchmarks for Optimization Software**, 1997.
http://plato.la.asu.edu/bench.html.

MURTAGH, B.A. and SAUNDERS, M.A., **MINOS 5.4 User's Guide**, Technical Report SOL 83-20R, December 1983. Revised January 1987, March 1993, February 1995, Systems Optimization Laboratory, Stanford University, Stanford, CA 94305-4022, 1995.

NASH, S.G., **Preconditioning of Truncated Newton Methods**, SIAM JOURNAL OF SCIENCE AND STATISTICAL COMPUTATIONS, Vol.6, 1985, pp.599-616.

NASH, S.G., POLYAK, R. and SOFER, A., **A Numerical Comparison of Barrier and Modified-barrier Methods for Large-scale Bound-constrained Optimization**, in W.W. Hager, D.W. Hearn and P.M. Pardalos (Eds.) Large Scale Optimization: State of the Art, KLUWER ACADEMIC PUBLISHERS, 1994, pp.319-338.

NOCEDAL, J., **Theory of Algorithms for Unconstrained Optimization**, ACTA NUMERICA, 1992.

ROSS, C., TERLAKY, T. and VIAL, J. PH., **Theory and Algorithms for Linear Optimization**, JOHN WILEY, 1997.

SCHITTKOWSKI, K., **More Test Examples for Nonlinear Programming Codes**, SPRINGER-VERLAG, New York, 1987.

SHANNO, D.F., BREITFELD, M.G. and SIMANTIRAKI, E.M., **Implementing Barrier Methods for Nonlinear Programming**, in T. Terlaky (Ed.) Interior Point Methods of Mathematical Programming, KLUWER ACADEMIC PUBLISHERS, Dordrecht, 1996, pp.399-414.

SHANNO, D.F. and SIMANTIRAKI, E.M., **Interior-point Methods for Linear and Nonlinear Programming**, in I.S. Duff and G.A. Watson (Eds.), The State of the Art in Numerical Analysis, OXFORD UNIVERSITY PRESS, New York, 1997, pp.339-362.

TAPIA, R., ZHANG, Y., SALTZMAN, M. and WEISER, A., **The Mehrotra Predictor-corrector Interior Point Method As A Perturbed Composite Newton Method**, SIAM JOURNAL ON OPTIMIZATION, Vol.6, No.1, 1996, pp.47-56.

VANDERBEI, R.J., **ALPO: Another Linear Program Optimizer**, Technical Report, AT&T Bell Laboratories, 1990.

VANDERBEI, R.J., **LOQO: An Interior Point Code for Quadratic Programming**, Technical Report SOR 94-15, Princeton University, 1994.

VANDERBEI, R.J., **Linear Programming. Foundations and Extensions**, KLUWER ACADEMIC PUBLISHERS, Boston, 1996.

VANDERBEI, R.J. and SHANNO, D.E., **An Interior-point Algorithm for Nonconvex Nonlinear Programming**, Technical Report SOR 97-21, Princeton University, 1997.

WOLFE, P., **Convergence Conditions for Ascent Methods**, SIAM REVIEW, Vol.11, 1969, pp.226-235.

WOLFE, P., **Convergence Conditions for Ascent Methods II: Some Corrections**, SIAM REVIEW, Vol.13, 1971, pp.185-188.

WRIGHT, M.H., **Interior Methods for Constrained Optimization**, in A. Iserles (Ed.), ACTA NUMERICA, 1992, Cambridge University Press, New York, 1992, pp.341-407.

WRIGHT, S.J., **Primal-Dual Interior-Point Methods**, SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS, Philadelphia, 1997.

WRIGHT, M.H., **The Interior-point Revolution in Constrained Optimization**, Technical Report 98-4-09, Bell Laboratories, Murray Hill, NJ 07974, June 30, 1998.

YE, Y., **Interior Point Algorithms. Theory and Analysis**, JOHN WILEY & SONS, New York, 1997.

ZOUTENDIJK, G., **Nonlinear Programming, Computational Methods**, in J.Abadie (Ed.) Integer and Nonlinear Programming, NORTH-HOLLAND, Amsterdam, 1970, pp.37-86.

APPENDIX

The starting points indicated in Table 11 are the following:

Problem	*Initial – Points*
HS32	I1=[1,7,2];I2=[9,9,2];I3=[9,9,9]
HS33	I1=[0,1,1]
HS37	I1=[30,30,30]
HS42	I1=[2,2,2,2];I2=[3,3,3,3];I3=[3,3,1,1]
HS43	I1=[1,0,0,0];I2=[1,1,0,0,];I3=[1,1,1,1]
HS44	I1=[1,2.5,0.5,3.5];I2=[1,4.5,0.5,3.5]
HS47	I1=[0.5,0.5,0.5,0.5,0.5];I2=[1,0.5,1,0.5,1]
HS48	I1=[3,5,-3,2,-1];I2=[1,7,2,-3,5]
HS49	I1=[9,8,1,-4,0.5]
HS50	I1=[10,-30,11,15,-15];I2=[30,-30,35,15,-35]; I3=[35,35,35,35,35];I4=[-35,-35,-35,-35,-35]
HS51	I1=[3,2.5,2,-1,2.5];I2=[-2,2.5,-2,-1,3]
HS52	I1=[20,20,20,20,20];I2=[-20,-20,-20,-20,-20] I3=[200,-200,200,-200,200]
HS53	I1=[10,10,10,10,10];I2=[-10,-10,-10,-10,-10] I3=[0,0,0,0,0]
HS61	I1=[3,-5,1]
HS63	I1=[3,3,3];I2=[1,1,1]
HS65	I1=[-5,4,0];I2=[5,4,0]
HS71	I1=[2,5,5,2]
HS73	I1=[2,1,1,0.5];I2=[2,2,2,0.5];I3=[2,2,2,2]
HS76	I1=[1,1,1,1];I2=[2,2,2,2];I3=[5,5,5,5]
HS113	I1=[10,3,1,1,1,2,4,3,6,10]; I2=[2,2,9,6,1,2,2,10,10,10]; I3=[20,20,20,20,20,20,20,20,20,20]
S217	I1=[1,1]
S219	I1=[1,1,1,1]