Predictor-Corrector Interior-Point Methods for Linear Constrained Optimization*

Neculai Andrei

Research Institute for Informatics
8-10, Averescu Avenue, Bucharest
ROMANIA
e-mail: nandrei@u3.ici.ro

Abstract: Predictor-corrector interior-point algorithms for linear constrained optimization problem are described. The algorithms are based on Mehrotra's predictor-corrector method, in the context of linear programming, and are modified to take into account the nonlinearity of the objective function. The convergence of the algorithms, as well as some other properties are proved. Computational results are provided for a number of linear constrained optimization problems in comparison with some codes such as TOLMIN (Powell), MINOS (Murtagh & Saunders), SPENBAR (Andrei), NLPQL (Schittkowski).

Keywords: linear programming, interior- point, predictor-corrector, linear optimization.

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

Since the publication of the article of Karmarkar [1984] an impressive number of papers has been written towards the convergence study of different variants of interior- point methods, and the implementation of the corresponding algorithms for solving the linear programming problem.

All these methods could be classified into three main categories: **projective-scaling** methods, **affine-scaling** methods and **path-following** methods. Projective scaling methods have been proposed by Karmarkar [1984] and studied by numerous researchers for the last decade. Affine scaling was originally considered by Dikin [1967] and further developed in two directions: **the range-space** and **the null-space** affine-scaling methods. Barnes [1986] and Vanderbei, Meketon and Freedman [1986] proposed a range-space version of the affine-scaling method in **primal** form. Adler, Karmarkar, Resende and Veiga [1986] proposed an affine-scaling method in **dual** form. Nazareth [1987] and Kim and Nazareth [1994] proposed and studied a null-space version of the affine-scaling method in primal form and tested its performance on a number of small relatively dense problems from the NETLIB collection [Gay, 1988]. A very important and significant advantage of the null-space version over the range-space version of the affine-scaling algorithms is that that the descent direction can be computed by an iterative method, i.e. it does not have to be computed exactly as the case is also with the projective-scaling methods.

The path-following algorithms start with the paper of Sonnevend [1985], who introduced the concept of **the analytic center of a polytope** as the unique point that minimizes the barrier function associated with that polytope. The analytic centers of all the constant-cost slices of the feasible set of the polytope determine the so- called **central path**. This path is a region with very attractive primal-dual properties,

**This research has been supported by the Romanian Academy through Grant No. 3649 / 16.07.1997. Special thanks are also extended to the Romanian Academy.*

and has been a key concept in the development of the path-following algorithms for linear programming. Karmarkar's algorithm and its variants were the first to implement the idea of staying near the central path leading to a polynomial complexity of $O(n^{3.5}L)$ operations, lower than Khachian's which was of $O(n^4L)$. (L is the total number of bits used in the description of the problem data.) Renegar [1988] followed the idea of staying near the central path and obtained the **first** real path-following algorithm with a complexity $O(\sqrt{n}L)$ lower than that of Karmarkar's method $O(nL)$ in terms of number of iterations. Later, Vaidya [1990] and Gonzaga [1991], following a penalty function approach, described algorithms with a complexity of $O(n^3L)$ operations, a limit that is still standing. Simultaneously, Kojima, Mizuno and Yoshise [1989a, 1989b], Monteiro and Adler [1989], Ye [1991], Freund [1991] and Gonzaga [1990, 1991, 1992] developed primal-dual path-following algorithms.

Recently, Lustig, Marsten and Shanno [1991,1992,1994], and Carpenter, Lustig, Mulvey and Shanno [1993] introduced a remarcable higher-order primal-dual and predictor-corrector logarithmic barrier methods for linear programming. They developed the theory of these methods and implemented the corresponding algorithms in some professional packages for linear programming (OB1 and OBN), and tested their performances on the NETLIB collection.

Some other efficient LP codes based on interior- point methods have been developed. Almost all of them are based on primal-dual or predictor-corrector algorithms, and they differ in many implementational aspects which are very important to consider. We can mention here the public domain research codes developed by:

Vanderbei - LoQo, written in C, which is an implementation of the predictor-corrector and primal-dual algorithms for LP and QP, being available from:
http://www.sor.princeton.edu/~rvdb/;

Zhang - LIPSOL, written in MATLAB and FORTRAN, also an implementation of the predictor-corrector method, available from:
http://pc5.math.umbc.edu/~yzhang/;

Gondzio - HOPDM, written in FORTRAN, which is an implementation of the higher- order primal-dual algorithm, available from:
http://ecolu-info.unige.ch/~logilab/software/;

Meszaros - BPMPD, written in FORTRAN, which is another implementation of the higher-order primal-dual algorithm, available from:
ftp://ftp.sztaki.hu/pub/oplab/SOFTWARE /BPMPD.

Being motivated by the performances of these interior- point path-following algorithms some researchers, notably Kortanek, Potra and Ye [1991], Goldfarb, Liu and Wang [1991], Wright [1992], and El-Bakry, Tapia, Tsuchiya and Zhang [1996] have extended these methods and algorithms to the nonlinear problems. In this paper we shall present an extension of the predictor-corrector approach to the linear optimization problems with convex, nonlinear objective functions.

2 Predictor-Corrector Interior-Point Algorithms

The problem we are concerned with is:

$$\begin{gathered} min f(x) \\ subject\ to: \\ Ax = b, \\ 0 \le x \le u, \end{gathered} \tag{2.1}$$

where $f(x) : R^n - R$ is a real, convex and at least continuously, two-differential function, $A \in R^{mxn}$ and some or all of the upper bounds u may be infinite.

In fact we can consider the problem in the following form:

$$\begin{gathered} min f(x) \\ subject\ to: \\ Ax = b, \\ x + s = u, \\ x \ge 0, s \ge 0, \end{gathered} \tag{2.2}$$

where $s \in R^n$ is the vector of slack variables. The predictor- corrector interior-point algorithm is motivated by applying the logarithmic barrier function to eliminate the inequality constraints in (2.2). This elimination can be done by incorporating them into the logarithmic barrier function, thus obtaining:

$$\begin{gathered} min f(x) - \mu \sum_{j=1}^{n} \ln x_j - \mu \sum_{j=1}^{n} \ln s_j \\ subject\ to: \\ Ax = b, \\ x + s = u. \end{gathered} \tag{2.3}$$

The Lagrangean function of (2.3) is:

$$\begin{gathered} L(x, s, y, w, \mu) = f(x) - \mu \sum_{j=1}^{n} \ln x_j - \\ \mu \sum_{j=1}^{n} \ln s_j - y^T(Ax - b) + w^T(x + s - u). \end{gathered} \tag{2.4}$$

The first-order necessary optimality conditions for a stationary point of (2.4) are:

$$\begin{aligned} Ax &= b, \\ x + s &= u, \\ \nabla f(x) - A^T y - z + w &= 0, \\ XZe &= \mu e, \\ SWe &= \mu e. \end{aligned} \tag{2.5}$$

where X, Z, S and W are diagonal matrices with the elements x_j, z_j, s_j and w_j respectively, and $z \in R^n$ is a vector of the dual slack variables.

The above optimality conditions express the primal and dual admissibility as well as the complementarity conditions: $x_j z_j = \mu$ and $s_j w_j = \mu, j = 1, ..., n$.

The idea of all interior- point algorithms is to solve the optimality conditions (2.5) by generating a sequence of strictly feasible solutions $\{[x\ s\ y\ z\ w]^T\}$ that converges to an optimal feasible pair $\{[x^*\ s^*\ y^*\ z^*\ w^*]^T\}$. The objective of such algorithms is to drive the duality gap to zero. This is implemented by solving the optimality conditions by means of the Newton method. According to the techniques used to solve the nonlinear system (2.5) the interior-point methods can be classified in:

1) Primal-Dual interior- point methods,

2) Predictor-Corrector interior- point methods,

3) Higher-order Predictor-Corrector interior-point methods.

For linear programming problems, the primal-dual interior- point methods were introduced by Megiddo [1989] using a logarithmic barrier function approach. Megiddo's idea was developed by Kojima, Mizuno and Yoshise [1989a, 1989b]. A different approach was proposed by Todd and Ye [1990]. This latter was based on reducing a primal-dual potential function that was similar to Karmarkar's primal potential function. Some other works on primal-dual interior -point algorithms are described in [Monteiro and Adler, 1989], [Lustig, 1988], [Gonzaga and Todd ,1992], [Choi, Monma and Shanno ,1990], [Lustig, Marsten and Shanno, 1990,1991,1992,1994], [Zhang, Tapia and Dennis, 1992], [Zhang and Tapia ,1993].

The predictor-corrector approach was introduced by Mizuno, Todd and Ye [1990] to describe a particular algorithm which considered a linear combination of a primal-dual affine step and a centered step. The predictor step is the primal-dual affine step studied by Monteiro, Adler and Resende [1990]. This step is then corrected by a centering step toward the central path, a concept introduced by Sonnevend [1985] and excellently presented by Gonzaga [1992]. The ideas have been extended and implemented in computer programs by Lustig, Marsten and Shanno [1992], Carpenter, Lustig, Mulvey and Shanno [1993], Vanderbei and Carpenter [1993], Potra [1996], Andersen, Gondzio, Meszaros and Xu [1996].

Mehrotra [1990,1992] introduced the higher-order predictor-corrector interior- point methods approach into the arsenal of the interior-point methods for linear programming. This follows from the composite Newton method, and has been studied and extended by Bayer and Lagarias [1989], Adler, Karmarkar, Resende and Veiga [1989a,1989b], Domich, Boggs, Rogers and Witzgall [1989], Carpenter, Lustig, Mulvey and Shanno [1993], Tapia, Zhang, Saltzman and Weiser [1996], Gondzio [1996], Andersen, Gondzio, Meszaros and Xu [1996], etc.

Thus, the predictor-corrector interior-point algorithm for (2.1) is derived from the first-order optimality conditions (2.5) by applying the Newton method (to find a solution to (2.5)) for a fixed, positive value of μ, and continuing this until the complementarity gap is reduced to a predetermined value.

Supposing that we have an estimation $t = [x\ s\ y\ z\ w]^T$ of the solution of (2.5) with $x > 0$, $s > 0$, $z > 0$ and $w > 0$, then we can compute the new point $t + \Delta t$, where $\Delta t = [\Delta x\ \Delta s\ \Delta y\ \Delta z\ \Delta w]^T$ is a solution of the following nonlinear algebraic system:

$$\begin{aligned} A(x + \Delta x) &= b, \\ (x + \Delta x) + (s + \Delta s) &= u, \\ A^T(y + \Delta y) + (z + \Delta z) - & \\ (w + \Delta w) &= \nabla f(x + \Delta x), \\ (X + \Delta X)(Z + \Delta Z)e &= \mu e, \\ (S + \Delta S)(W + \Delta W)e &= \mu e, \end{aligned} \tag{2.6}$$

where $\Delta X, \Delta S, \Delta Z, \Delta W$ are diagonal matrices with elements Δx_j, Δs_j, Δz_j, and respectively Δw_j (j=1,...,n).

Considering the objective $f(x)$ a convex, continuously two-differential function, from (2.6) we obtain:

$$\begin{aligned} A\Delta x &= b - Ax, \\ \Delta x + \Delta s &= u - x - s, \\ -\nabla^2 f(x)\Delta x + A^T\Delta y + \Delta z - \Delta w &= \\ \nabla f(x) - A^T y + w - z & \\ X\Delta z + Z\Delta x &= \mu e - XZe - \Delta X\Delta Ze, \\ S\Delta w + W\Delta s &= \mu e - SWe - \Delta S\Delta We. \end{aligned} \tag{2.7}$$

As we can easily see, the great difference between (2.7) and the corresponding system for the linear programming case is the presence of the Hessian matrix $\nabla^2 f(x)$ of the objective function.

As in the case of linear programming, the idea of the predictor-corrector interior-point method is to solve the system (2.7) in two stages, [Mehrotra, 1990].

In the first stage, the so- called **prediction stage**, we consider $\mu = 0$ and ignore the quadratical terms $\Delta X \Delta Z e$, and $\Delta S \Delta W e$,thus solving the linear system:

$$\begin{aligned} A\Delta x &= b - Ax, \\ \Delta x + \Delta s &= u - x - s, \\ -\nabla^2 f(x)\Delta x + A^T \Delta y + \Delta z - \Delta w &= \\ \nabla f(x) - A^T y + w - z & \qquad (2.8) \\ X\Delta z + Z\Delta x &= -XZe, \\ S\Delta w + W\Delta s &= -SWe, \end{aligned}$$

subject to the **primal-dual affine direction**: $\Delta t_a = [\Delta x_a \ \Delta s_a \ \Delta y_a \ \Delta z_a \ \Delta w_a]^T$.

These directions are then used for two distinct purposes: to approximate the quadratical terms in the right-hand side terms of (2.7), and to dynamically estimate the barrier parameter μ.

For linear programming problem, to estimate μ, some authors (Mehrotra [1990, 1992], Lustig, Marsten and Shanno [1991,1992,1994], Carpenter, Lustig, Mulvey and Shanno [1993]) perform the standard ratio test on both the primal and the dual variables in order to determine the step that would actually be taken if the primal-dual affine direction given by (2.8) were used. For linear constrained optimization problems we consider the same strategy, thus defining

$$\begin{aligned} \overset{\wedge}{\delta}_p = \min\{\min_{j=1,n}\{\tfrac{x_j}{-\Delta x_{aj}}, \Delta x_{aj} < 0\}, \\ \min_{j=1,n}\{\tfrac{s_j}{-\Delta s_{aj}}, \Delta s_{aj} < 0\}\}, \end{aligned}$$

and (2.9)

$$\begin{aligned} \overset{\wedge}{\delta}_d = \min\{\min_{j=1,n}\{\tfrac{z_j}{-\Delta z_{aj}}, \Delta z_{aj} < 0\}, \\ \min_{j=1,n}\{\tfrac{w_j}{-\Delta w_{aj}}, \Delta w_{aj} < 0\}\}, \end{aligned}$$

and considering

$$\begin{aligned} \delta_p &= \tau \overset{\wedge}{\delta}_p, \\ \delta_d &= \tau \overset{\wedge}{\delta}_d . \end{aligned}$$

where $\tau \in (0, 1)$, (usually $\tau = 0.99995$).

Proposition 1. The duality gap corresponding to the current solution is:

$$g = x^T z + s^T w. \qquad (2.10)$$

Proof. For the problem (2.1) and its dual, the duality gap is $g = \nabla f(x)^T x - b^T y + u^T w$. From (2.5) we have: $g = (y^T A + z^T - w^T)x - b^T y + u^T w = z^T x + u^T w - x^T w = x^T z + s^T w$.

The duality gap which would result from a step in the affine direction is :

$$\begin{aligned} \hat{g} = (x + \delta_p \Delta x_a)^T (z + \delta_d \Delta z_a) + \\ (s + \delta_p \Delta s_a)^T (w + \delta_d \Delta w_a). \qquad (2.11) \end{aligned}$$

With these simple algebraic constructions, an estimation of the barrier parameter μ is then:

$$\mu = \begin{cases} \sigma\left(\frac{\hat{g}}{x^T z + s^T w}\right)^2 \left(\frac{\hat{g}}{n}\right), & if \ x^T z + s^T w \geq 1, \\ \sigma \frac{x^T z + s^T w}{n^2}, & if \ x^T z + s^T w < 1, \end{cases} \qquad (2.12)$$

where $\sigma \in (0, 1)$ is a parameter for modifying the value of μ (usually $\sigma = 0.1$).

With this value of μ we can initiate the second stage, the so -called **correction (centering) stage**, in which the following linear algebraic system is solved:

$$\begin{aligned} A\Delta x &= 0, \\ \Delta x + \Delta s &= 0, \qquad (2.13) \\ -\nabla^2 f(x)\Delta x + A^T \Delta y + \Delta z - \Delta w &= 0, \\ X\Delta z + Z\Delta x &= \mu e - \Delta X_a \Delta Z_a e, \\ S\Delta w + W\Delta s &= \mu e - \Delta S_a \Delta W_a e, \end{aligned}$$

subject to **the centering direction** $\Delta t_c = [\Delta x_c \ \Delta s_c \ \Delta y_c \ \Delta z_c \ \Delta w_c]^T$, thereby we can compute the actual new direction (the so -called predictor- corrector direction) $\Delta t = \Delta t_a + \Delta t_c$.

Alternatively, we can compute the **full** direction Δt **directly** by solving the linear system:

$$\begin{aligned} A\Delta x &= b - Ax, \\ \Delta x + \Delta s &= u - x - s, \\ -\nabla^2 f(x)\Delta x + A^T \Delta y + \Delta z - \Delta w &= \\ \nabla f(x) - A^T y + w - z & \qquad (2.14) \\ X\Delta z + Z\Delta x &= \mu e - XZe - \Delta X_a \Delta Z_a e, \\ S\Delta w + W\Delta s &= \mu e - SWe - \Delta S_a \Delta W_a e, \end{aligned}$$

for Δx, Δs, Δy, Δz and Δw. We notice that the system (2.14) includes the correction terms $\Delta X_a \Delta Z_a e$, and $\Delta S_a \Delta W_a e$ in the right-hand side of the Newton system.

Clearly, the difference between the two algebraic systems (2.8) and (2.13), or (2.14) is the right-hand side term, so the matrix algebra for solving (2.13) mainly remains the same as in the solution of (2.8).

With this new direction Δt we perform the ratio tests to determine the actual step sizes $\hat{\alpha}_p$ and $\hat{\alpha}_d$:

$$\hat{\alpha}_p = \min\{\min_{j=1,n}\{\tfrac{x_j}{-\Delta x_j}, \Delta x_j < 0\},$$
$$\min_{j=1,n}\{\tfrac{s_j}{-\Delta s_j}, \Delta s_j < 0\}\},$$

and (2.15)

$$\hat{\alpha}_d = \min\{\min_{j=1,n}\{\tfrac{z_j}{-\Delta z_j}, \Delta z_j < 0\},$$
$$\min_{j=1,n}\{\tfrac{w_j}{-\Delta w_j}, \Delta w_j < 0\}\},$$

and then the actual new point is computed as:

$$\begin{aligned}\tilde{x} &= x + \hat{\alpha}_p\, \Delta x,\\ \tilde{s} &= s + \hat{\alpha}_p\, \Delta s,\\ \tilde{y} &= y + \hat{\alpha}_d\, \Delta y,\\ \tilde{z} &= z + \hat{\alpha}_d\, \Delta z,\\ \tilde{w} &= w + \hat{\alpha}_d\, \Delta w.\end{aligned} \tag{2.16}$$

Given a current estimate to the optimal point that is primal and dual feasible, the direction Δt is a **feasible direction** if satisfying:

$$\begin{aligned}A\Delta x &= 0,\\ \Delta x + \Delta s &= 0,\\ -\nabla^2 f(x)\Delta x + A^t\Delta y + \Delta z - \Delta w &= 0.\end{aligned} \tag{2.17}$$

Proposition 2. For any $\Delta t = [\Delta x\ \Delta s\ \Delta y\ \Delta z\ \Delta w]^T$ that satisfy (2.17). $\Delta x^T\Delta z + \Delta s^T\Delta w = \Delta x^T\nabla^2 f(x)\Delta x$.

Proof. Using (2.17) we have: $\Delta z = \nabla^2 f(x)\Delta x + \Delta w - A^T\Delta y$, and $\Delta x = -\Delta s$. Hence, $\Delta x^T\Delta z + \Delta s^T\Delta w = \Delta x^T(\nabla^2 f(x)\Delta x + \Delta w - A^T\Delta y) + \Delta s^T\Delta w =$
$\Delta x^T\nabla^2 f(x)\Delta x - \Delta s^T\Delta w - (A\Delta x)^T\Delta y + \Delta s^T\Delta w = \Delta x^T\nabla^2 f(x)\Delta x$.

For the predictor-corrector method, which we are considering here, assuming the same step-lengths are taken in the primal and the dual subspaces. the change in complementarity is the following:

$$\begin{gathered}(x+\alpha(\Delta x_a+\Delta x_c))^T(z+\alpha(\Delta z_a+\Delta z_c))+\\ (s+\alpha(\Delta s_a+\Delta s_c))^T(w+\alpha(\Delta w_a+\Delta w_c)) =\\ x^T z + s^T w+\\ \alpha(x^T\Delta z_a + z^T\Delta x_a + x^T\Delta z_c + z^T\Delta x_c+\\ s^T\Delta w_a + w^T\Delta s_a + s^T\Delta w_c + w^T\Delta s_c)+\\ \alpha^2((\Delta x_a+\Delta x_c)^T(\Delta z_a+\Delta z_c)+(\Delta s_a+\\ \Delta s_c)^T(\Delta w_a+\Delta w_c) =\\ x^T z + s^T w - \alpha(x^T z + s^T w) - \alpha(\Delta x_a^T\Delta z_a+\\ \Delta s_a^T\Delta w_a - 2n\mu)+\\ \alpha^2((\Delta x_a+\Delta x_c)^T(\Delta z_a+\Delta z_c)+(\Delta s_a+\\ \Delta s_c)^T(\Delta w_a+\Delta w_c) =\\ x^T z + s^T w - \alpha(x^T z + s^T w+\\ \Delta x_a^T\nabla^2 f(x)\Delta x_a - 2n\mu)+\\ \alpha^2((\Delta x_a+\Delta x_c)^T(\Delta z_a+\Delta z_c)+(\Delta s_a+\\ \Delta s_c)^T(\Delta w_a+\Delta w_c)\end{gathered}$$

by proposition 2. Since $(\Delta t_a + \Delta t_c)$ is itself a feasible direction, by proposition 2 we obtain:

$$\begin{gathered}(x+\alpha(\Delta x_a+\Delta x_c))^T(z+\alpha(\Delta z_a+\Delta z_c))+\\ (s+\alpha(\Delta s_a+\Delta s_c))^T(w+\alpha(\Delta w_a+\Delta w_c)) =\\ x^T z + s^T w - \alpha(x^T z + s^T w+\\ \Delta x_a^T\nabla^2 f(x)\Delta x_a - 2n\mu)+\\ \alpha^2((\Delta x_a+\Delta x_c)^T\nabla^2 f(x)(\Delta x_a+\Delta x_c)).\end{gathered}$$

In contrast with the primal-dual approach, for the predictor- corrector method, although there is a second-order term apparently increasing the complementarity, there is now a new first-order term, $\Delta x_a^T\nabla^2 f(x)\Delta x_a$,reducing the complementarity. Thus, for the case of small step-length, the predictor-corrector method has better complementarity reduction than the primal-dual one. Having in mind that $\alpha \in (0,1)$ this proves the convergence of the method.

2.1 The Predictor-Corrector Algorithm

The predictor-corrector interior-point algorithm can now be described quite simply. It involves the solving at each iteration of two linear algebraic systems with the same coefficient matrix, but different right-hand side terms. as well as some other simple algebraic manipulations.

Algorithm PCLC

Step1. Choose an initial approximation $t_0 = [x_0\ s_0\ y_0\ z_0\ w_0]^T$ for which $x_0 > 0$, $s_0 > 0$, $z_0 > 0$, and $w_0 > 0$. Consider $\sigma = 0.1$ and $k = 0$.

Step 2. Test the relative duality gap:

$$\frac{|\nabla f(x_k)^T x_k - b^T y_k + u^T w_k|}{1+|f(x_k) - \nabla f(x_k)^T x_k + b^T y_k - u^T w_k|} \le \varepsilon.$$

Step 3. With this estimation of variables, solve the linear algebraic system (2.8) subject to the affine direction Δt_a.

Step 4. Consider $\sigma_k = \min(\sigma, x_k^T z_k + s_k^T w_k)$. Compute the barrier parameter μ as in (2.12), where σ is replaced by the value of σ_k.

Step 5. Solve the linear algebraic system (2.14) for the **full** direction $\Delta t = [\Delta x\ \Delta s\ \Delta y\ \Delta z\ \Delta w]^T$.

Step 6. Choose $\tau \in (0,1)$ and consider $\alpha_p = \tau\,\hat{\alpha}_p$ and $\alpha_d = \tau\,\hat{\alpha}_d$, where $\hat{\alpha}_p$ and $\hat{\alpha}_d$ are given by (2.15).

Step 7. Consider the new estimation t_{k+1} of the solution:

$$\begin{aligned} x_{k+1} &= x_k + \alpha_p \Delta x, \\ s_{k+1} &= s_k + \alpha_p \Delta s, \\ y_{k+1} &= y_k + \alpha_d \Delta y, \\ z_{k+1} &= z_k + \alpha_d \Delta z, \\ w_{k+1} &= w_k + \alpha_d \Delta w, \end{aligned}$$

set $k = k + 1$, and continue with step 2.

Some remarks are in order:

1. The systems we solve to determine the predictor, the corrector, or the full predictor-corrector direction involve the same matrix. That is, each of these directions is obtained based on the same factorization of the matrix.

2. At each iteration of the algorithm we obtain an admissible solution which satisfies the primal and the dual constraints. The only motivation for continuing the iterations is to reduce the duality gap.

3. In contrast with linear programming problems, in this case the corresponding Newton systems (2.8) and (2.14) involve the Hessian $\nabla^2 f(x)$ of the objective function. This will introduce some complications to be further considered.

Example 1. Let us consider the following problem:

$$\min[x_1^2 + \tfrac{1}{2}x_2^2 + x_3^2 + \tfrac{1}{2}x_4^2 - x_1x_3 + x_3x_4 - x_1 - \\ 3x_2 + x_3 - x_4 + \\ e^{x_1} + e^{x_2} + e^{x_3} + e^{x_4}]$$

subject to:

$$\begin{gathered} x_1 + 2x_2 + x_3 + x_4 + x_5 = 5, \\ 3x_1 + x_2 + 2x_3 - x_4 + x_6 = 4, \\ x_2 + 4x_3 - x_7 = 1.5, \\ 0 \le x_i \le 10, (i = 1, ..., 7). \end{gathered}$$

The PCLC algorithm, initialized at point: $x_0 = [0.5...0.5]^T, s_0 = [1...1]^T, y_0 = [111]^T, z_0 = [1...1]^T, w_0 = [1...1]^T$ gives the following solution:

x_1	0.04421
x_2	0.9654
x_3	0.1336
x_4	0.0000
x_5	2.8912
x_6	2.6346
x_7	0.0000

Considering: $\sigma = 0.1$, $\tau_k = 0.99995$, and $\varepsilon = 10^{-8}$, the characteristics of the optimization process are presented in the Table:

No.	FobP	FobD
1	4.0167	562.9565
2	8.1241	1955.436
3	6.8792	2580.719
4	6.6720	4182.258
5	5.3038	51.4250
6	3.6913	3.6800
7	3.5014	3.4820
8	3.4887	3.4870
9	3.4872	3.4871
10	3.4871	3.4871

μ	g	α
0.4098	10.5000	0.1149
12.3577	110.9070	0.1895
4.8041	645.8158	0.1458
$0.3451e^{-2}$	1051.835	1.5102
$0.1104e^{-1}$	4631.276	1.0370
$0.6004e^{-2}$	42.4592	0.9972
$0.9916e^{-3}$	0.4859	1.0137
$0.1479e^{-4}$	$0.2692e^{-1}$	1.0115
$0.6625e^{-7}$	$0.1801e^{-2}$	1.0035
$0.2484e^{-10}$	$0.3488e^{-4}$	1.0000

$g = x_k^T z_k + s_k^T w_k$.(duality gap)
$\alpha = \min\{\alpha_p, \alpha_d\}$.(step length)

As one can see, the algorithm needs a small number of iterations, and then of functions evaluations. At each iteration the current solution is primal and dual feasible. The algorithm continues to execute iterations in order to reduce the complementarity. The value of the barrier parameter μ has a nonmonotonous evolution. This is typical for the interior- point methods, being a heritage from the penalty methods of Fiacco and McCormick [1968]. The step- length along the full direction of minimization is going to 1, exactly as in the Newton method.

2.2 The Multiple Predictor-Corrector Algorithm

As already noticed, the process of solving the necessary optimality conditions given by the nonlinear system (2.5) is split into the solving of the system (2.8) for the affine direction and the solving of the system (2.13) for the correction direction. It is very likely that the affine direction Δt_a, defined at step 3 of the PCLC algorithm, points to the boundary of the positive orthant, thus involving a very small step-length. The role of the centering step Δt_c,defined by the system (2.13) is to redress this situation. As in the linear programming case, we can consider the **composite Newton method** for the determination of the correction direction, thus obtaining a **higher-order predictor- corrector algorithm**.

Considering the optimality conditions (2.5), let us define the following nonlinear transformation:

$$F(t) = \begin{bmatrix} Ax - b \\ x + s - u \\ A^T y + z - w - \nabla f(x) \\ XZe \\ SWe \end{bmatrix} \qquad (2.18)$$

Then the PCLC algorithm at step 3 solves the linear algebraic system:

$$F'(t_k)\Delta t_a = -F(t_k), \qquad (2.19)$$

subject to the affine direction Δt_a. At step 5 the PCLC algorithm solves the following linear algebraic system:

$$F'(t_k)\Delta t = -F(t_k) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mu - \Delta X_a \Delta Z_a e \\ \mu - \Delta S_a \Delta W_a e \end{bmatrix} \qquad (2.20)$$

subject to the full direction Δt.

The higher-order predictor method considers the same strategy in determining the affine direction, but the composite Newton method for the full direction. Thus, **the level m_k composite Newton interior -point method**, at iteration k, instead of solving one linear system (2.20) for the full direction, will solve a number of m_k linear systems with different right-hand side terms of the following form:

$$F'(t_k)\Delta t_i = -F(t_k + \sum_{j=0}^{i-1} \Delta t_j) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mu - \Delta X_a \Delta Z_a e \\ \mu - \Delta S_a \Delta W_a e \end{bmatrix} \qquad (2.21)$$

for i=1,....m_k, where $\Delta t_0 = \Delta t_a$, and define

$$\Delta t = \sum_{j=0}^{m_k} \Delta t_j .$$

Observe that at each iteration of the corresponding algorithm it is necessary to solve $m_k + 1$ linear algebraic systems with the same coefficient matrix. After the coefficient matrix has been factored, the additional work of the composite predictor-corrector method is in the extra backsolve to compute the full direction. That what is gained from this extra work is a better approximation of the centering direction.

Thus we can derive a **level-m composite Newton interior- point algorithm** which is the same as PCLC in which at step 5 a number of m_k (2.21) linear systems are solved for the full direction.

Theorem 1. The predictor-corrector interior- point algorithm PCLC is equivalent to the level-1 composite Newton interior -point algorithm.

Proof. This follows immediately from the above discussions.

Instead of pursuing this strategy we can use another one which could be considered as a generalization of the PCLC algorithm in a **multiple predictor-corrector procedure.** Instead of solving the correcting system (2.13) once at each iteration of the predictor-corrector interior-point method, it can be solved repetitively with each direction corrected based on the previous direction. The number of corrections m_k in an iteration k is dynamically chosen according to the reduction of the duality gap.

Algorithm MPCLC

Step1. Choose an initial approximation $t_0 = [x_0 \ s_0 \ y_0 \ z_0 \ w_0]^T$ for which $x_0 > 0$, $s_0 > 0$, $z_0 > 0$, and $w_0 > 0$. Consider $\sigma = 0.1$ and $k = 0$.

Step 2. Test the relative duality gap:

$$\frac{\left|\nabla f(x_k)^T x_k - b^T y_k + u^T w_k\right|}{1 + \left|f(x_k) - \nabla f(x_k)^T x_k + b^T y_k - u^T w_k\right|} \leq \varepsilon .$$

Step 3. With this estimation of variables, solve the linear algebraic system (2.8) subject to the affine direction Δt_a.

Step 4. Consider $\sigma_k = \min(\sigma, x_k^T z_k + s_k^T w_k)$. Compute the barrier parameter μ as in (2.12), where σ is replaced by the value of σ_k.

Step 5. For $i = 1, ..., m_k$ solve the following linear system for $\Delta t^i = \left[\Delta x^i \ \Delta s^i \ \Delta y^i \ \Delta z^i \ \Delta w^i\right]^T$:

$$\begin{aligned} A\Delta x^i &= b - Ax \\ \Delta x^i + \Delta s^i &= u - x - s \\ -\nabla^2 f(x)\Delta x^i + A^T \Delta y^i + \Delta z^i - \Delta w^i &= \end{aligned} \quad (2.22)$$

$$\nabla f(x) - A^T y + w - z$$

$$X\Delta z^i + Z\Delta x^i = \mu e - XZe - \Delta X^{i-1}\Delta Z^{i-1}e$$

$$S\Delta w^i + W\Delta s^i = \mu e - SWe - \Delta S^{i-1}\Delta W^{i-1}e$$

where $\Delta X^0 = \Delta X_a, \Delta Z^0 = \Delta Z_a, \Delta S^0 = \Delta S_a$ and $\Delta W^0 = \Delta W_a$.

Define $\Delta t = \Delta t^{m_k}$.

Step 6. Choose $\tau \in (0,1)$ and consider $\alpha_p = \tau\,\hat{\alpha}_p$ and $\alpha_d = \tau\,\hat{\alpha}_d$, where $\hat{\alpha}_p$ and $\hat{\alpha}_d$ are given by (2.15).

Step 7. Consider the new estimation t_{k+1} of the solution:

$$\begin{aligned} x_{k+1} &= x_k + \alpha_p \Delta x, \\ s_{k+1} &= s_k + \alpha_p \Delta s, \\ y_{k+1} &= y_k + \alpha_d \Delta y, \\ z_{k+1} &= z_k + \alpha_d \Delta z, \\ w_{k+1} &= w_k + \alpha_d \Delta w, \end{aligned}$$

set $k = k + 1$, and continue with step 2.

Observe again that in the MPCLC algorithm, after the coefficient matrix of the system (2.8) has been factored, the additional algebra is in some extra backsolve to compute the full direction.

Example 2. Let us consider the same problem as in Example 1 above. Considering the same initialization, the MPCLC algorithm yields the same solution, but the characteristics of the optimization process are illustrated in the Table below:

No	FobP	FobD	μ
1	5.5787	-0.5044	0.4098
2	5.3408	7.7489	0.0463
3	4.7246	7.3848	$0.3847e^{-2}$
4	4.0469	5.8586	$0.5971e^{-3}$
5	3.4910	3.4285	$0.1307e^{-2}$
6	3.4871	3.4871	$0.6993e^{-5}$
7	3.4871	3.4871	$0.9767e^{-10}$

g	α	m_k
10.500	0.1005	1
4.4531	0.1868	1
2.2778	0.3008	1
1.3947	0.4714	1
0.6406	1.0081	10
$0.1851e^{-1}$	1.0018	7
$0.6918e^{-4}$	0.9995	2

$g = x_k^T z_k + s_k^T w_k$.(duality gap)
$\alpha = \min\{\alpha_p, \alpha_d\}$.(step -length)
m_k = number of corrections at iteration k.

As one can see, the number of centering corrections is relatively small. This is in line with the theory of the composite Newton method, as well as with the numerical experiments from the linear programming case [Ortega and Rheinboldt, 1970], [Tapia, Zhang, Saltzman and Weiser, 1996].

Theorem 2. If $m_k = 1$ for all k then the multiple predictor-corrector algorithm (MPCLC) is equivalent to the predictor-corrector algorithm (PCLC).

Proof. Considering $m_k = 1$ in algorithm MPCLC (step 5), we must solve the system (2.22) which is the same linear system as (2.14).

The following Theorem is a specialization of Theorem 2.1 of Carpenter, Lustig, Mulvey and Shanno [1993].

Theorem 3. The multiple predictor-corrector algorithm (MPCLC) is equivalent to the composite Newton interior - point method.

To prove the equivalence of these two algorithms, notice that they are all identical, except for step 5, then the proof of the Theorem requires only that the direction Δt given by MPCLC is the same as that given by the composite Newton interior- point method. Before proceeding with the proof of the Theorem we need to prove some propositions.

Proposition 3. For any $i \geq 1$ of the composite Newton method, we have:

$$b - A(x_k + \sum_{j=0}^{i} \Delta x^j) = 0,$$

$$u - (x_k + \sum_{j=0}^{i} \Delta x^j) - (s_k + \sum_{j=0}^{i} \Delta s^j) = 0,$$

$$\nabla f(x) - A^T(y_k + \sum_{j=0}^{i} \Delta y^j) - (z_k + \sum_{j=0}^{i} \Delta z^j) + (w_k + \sum_{j=0}^{i} \Delta w^j) = 0.$$

Proof. From the definition of $F(t)$ in (2.18) we have that

$$A\Delta x^i = b - A(x_k + \sum_{j=0}^{i-1} \Delta x^j),$$

$$\Delta x^i + \Delta s^i = u - (x_k + \sum_{j=0}^{i-1} \Delta x^j) - (s_k + \sum_{j=0}^{i-1} \Delta s^j),$$

$$A^T \Delta y^i + \Delta z^i - \Delta w^i = \nabla f(x) - A^T(y_k + \sum_{j=0}^{i-1} \Delta y^j) - (z_k + \sum_{j=0}^{i-1} \Delta z^j) + (w_k + \sum_{j=0}^{i-1} \Delta w^j).$$

which immediately implies the result of the proposition.

Proposition 4. For any $i \geq 0$ of the composite Newton method, we have:

$$(X^k + \sum_{j=0}^{i} \Delta X^j)(Z^k + \sum_{j=0}^{i} \Delta Z^j)e = \mu e +$$
$$(\sum_{j=0}^{i} \Delta X^j)(\sum_{j=0}^{i} \Delta Z^j)e - (\sum_{j=0}^{i-1} \Delta X^j)(\sum_{j=0}^{i-1} \Delta Z^j)e,$$
$$(S^k + \sum_{j=0}^{i} \Delta S^j)(W^k + \sum_{j=0}^{i} \Delta W^j)e = \mu e +$$
$$(\sum_{j=0}^{i} \Delta S^j)(\sum_{j=0}^{i} \Delta W^j)e - (\sum_{j=0}^{i-1} \Delta S^j)(\sum_{j=0}^{i-1} \Delta W^j)e.$$

Proof. Considering the first relation and multiplying term by term, we have:

$$(X^k + \sum_{j=0}^{i} \Delta X^j)(Z^k + \sum_{j=0}^{i} \Delta Z^j)e =$$
$$X^k Z^k e + \sum_{j=0}^{i} (X^k \Delta Z^j + Z^k \Delta X^j)e +$$
$$(\sum_{j=0}^{i} \Delta X^j)(\sum_{j=0}^{i} \Delta Z^j)e. \quad (2.23)$$

But, from the composite Newton interior-point method we have that:

$$(X^k \Delta Z^i + Z^k \Delta X^i) = \mu e -$$
$$(X^k + \sum_{j=0}^{i-1} \Delta X^j)(Z^k + \sum_{j=0}^{i-1} \Delta Z^j)e, \quad (2.24)$$

which implies that:

$$X^k Z^k e + \sum_{j=0}^{i} (X^k \Delta Z^j + Z^k \Delta X^j)e +$$
$$(\sum_{j=0}^{i-1} \Delta X^j)(\sum_{j=0}^{i-1} \Delta Z^j)e = \mu e. \quad (2.25)$$

Now, adding and subtracting in the right-hand side of Equation (2.23) the term $(\sum_{j=0}^{i-1} \Delta X^j)(\sum_{j=0}^{i-1} \Delta Z^j)e$ we get:

$$(X^k + \sum_{j=0}^{i} \Delta X^j)(Z^k + \sum_{j=0}^{i} \Delta Z^j)e =$$
$$X^k Z^k e + \sum_{j=0}^{i} (X^k \Delta Z^j + Z^k \Delta X^j)e +$$
$$(\sum_{j=0}^{i-1} \Delta X^j)(\sum_{j=0}^{i-1} \Delta Z^j)e + (\sum_{j=0}^{i} \Delta X^j)$$
$$(\sum_{j=0}^{i} \Delta Z^j)e - (\sum_{j=0}^{i-1} \Delta X^j)(\sum_{j=0}^{i-1} \Delta Z^j)e.$$

Using (2.25) for substituting μe for the first three terms we get the result of the proposition. Applying the same approach to the second relation in S and W matrices, we obtain the complete proof of the proposition.

Now if combining the results of the propositions 3 and 4 we have that

$$F(t^k + \sum_{j=0}^{i} \Delta t^j) =$$
$$\begin{bmatrix} b - A(x_k + \sum_{j=0}^{i} \Delta x^j) \\ u - (x_k + \sum_{j=0}^{i} \Delta x^j) - \\ (s_k + \sum_{j=0}^{i} \Delta s^j) \\ \nabla f(x) - A^T(y_k + \sum_{j=0}^{i} \Delta y^j) - \\ (z_k + \sum_{j=0}^{i} \Delta z^j) + \\ (w_k + \sum_{j=0}^{i} \Delta w^j) \\ (X^k + \sum_{j=0}^{i} \Delta X^j)(Z^k + \sum_{j=0}^{i} \Delta Z^j)e \\ (S^k + \sum_{j=0}^{i} \Delta S^j)(W^k + \sum_{j=0}^{i} \Delta W^j)e \end{bmatrix} \quad (2.26)$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mu e + (\sum_{j=0}^{i} \Delta X^j)(\sum_{j=0}^{i} \Delta Z^j)e - \\ (\sum_{j=0}^{i-1} \Delta X^j)(\sum_{j=0}^{i-1} \Delta Z^j)e \\ \mu e + (\sum_{j=0}^{i} \Delta S^j)(\sum_{j=0}^{i} \Delta W^j)e - \\ (\sum_{j=0}^{i-1} \Delta S^j)(\sum_{j=0}^{i-1} \Delta W^j)e \end{bmatrix} \quad (2.27)$$

To prove Theorem 3 we must prove that the direction $\Delta t = \sum_{i=0}^{m_k} \Delta t^i$ is the same as the direction Δt^{m_k} given by the MPCLC algorithm.

To do this, we note that Δt solves the following system:

$$A \Delta x = b - Ax,$$
$$\Delta x + \Delta s = u - x - s,$$
$$A^T \Delta y + \Delta z - \Delta w = \nabla f(x) - A^T y + w - z,$$
$$Z \Delta x + X \Delta z = \mu e - XZe -$$
$$(\sum_{i=0}^{m_k - 1} \Delta X^i)(\sum_{i=0}^{m_k - 1} \Delta Z^i)e,$$

$$W\Delta s + S\Delta w = \mu e - SWe - \\ (\sum_{i=0}^{m_k-1} \Delta S^i)(\sum_{i=0}^{m_k-1} \Delta W^i)e.$$

Thus, Δt^{m_k} and Δt solve precisely the same system when $(\sum_{i=0}^{m_k-1} \Delta t^i) = \Delta t^{m_k-1}$.

Proposition 5. For all $p \geq 0$ we have that $(\sum_{i=0}^{p} \Delta t^i) = \Delta t^p$.

Proof (by induction) For $p = 0$, the result of proposition is true; Δt^0 given by the two algorithms is the affine direction.

Assume that the proposition is true for $1 \leq p \leq m-1$. We must prove that it is true for $p = m$. By adding all the systems defining each Δt^i we have that $\sum_{i=0}^{m} \Delta t^i$ solves the system:

$$-F(t)+ \sum_{i=0}^{m-1} [\mu \hat{e} - F(t+ \sum_{j=0}^{i} \Delta t^j)] = \left[\begin{array}{c} A\sum_{i=0}^{m} \Delta x^i \\ \sum_{i=0}^{m} \Delta x^i + \sum_{i=0}^{m} \Delta s^i \\ A^T(\sum_{i=0}^{m} \Delta y^i) + \sum_{i=0}^{m} \Delta z^i - \sum_{i=0}^{m} \Delta w^i \\ Z(\sum_{i=0}^{m} \Delta x^i) + X(\sum_{i=0}^{m} \Delta z^i) \\ W(\sum_{i=0}^{m} \Delta s^i) + S(\sum_{i=0}^{m} \Delta w^i) \end{array}\right] \tag{2.28}$$

where $\hat{e}$ is a vector with 1 in the last 2n components and 0 otherwise.

Replacing $F(t+ \sum_{j=0}^{i} \Delta t^j)$ from (2.26) in (2.28) we get:

$$-F(t)+ \sum_{i=0}^{m-1} [\mu \hat{e} - F(t+ \sum_{j=0}^{i} \Delta t^j)] =$$
$$\left[\begin{array}{c} b - Ax \\ u - x - s \\ \nabla f(x) - A^T y - z + w \\ \mu e - XZe - \Delta X^0 \Delta Z^0 e + \{ \sum_{i=1}^{m-1} (\mu e - \\ [\mu e + (\sum_{j=0}^{i} \Delta X^j)(\sum_{j=0}^{i} \Delta Z^j)e - \\ (\sum_{j=0}^{i-1} \Delta X^j)(\sum_{j=0}^{i-1} \Delta Z^j)e])\} \\ \mu e - SWe - \Delta S^0 \Delta W^0 e + \{ \sum_{i=1}^{m-1} (\mu e - \\ [\mu e + (\sum_{j=0}^{i} \Delta S^j)(\sum_{j=0}^{i} \Delta W^j)e - \\ (\sum_{j=0}^{i-1} \Delta S^j)(\sum_{j=0}^{i-1} \Delta W^j)e])\} \end{array}\right]$$

Taking into consideration the above relations (2.25), within the braces the μ terms cancel as do the alternating terms in the summation over i. This implies

$$-F(t)+ \sum_{i=0}^{m-1} [\mu \hat{e} - F(t+ \sum_{j=0}^{i} \Delta t^j)] =$$
$$\left[\begin{array}{c} b - Ax \\ u - x - s \\ \nabla f(x) - A^T y - z + w \\ \mu e - XZe - (\sum_{i=0}^{m-1} \Delta X^i)(\sum_{i=0}^{m-1} \Delta Z^i)e \\ \mu e - SWe - (\sum_{i=0}^{m-1} \Delta S^i)(\sum_{i=0}^{m-1} \Delta W^i)e \end{array}\right] = \left[\begin{array}{c} b - Ax \\ u - x - s \\ \nabla f(x) - A^T y - z + w \\ \mu e - XZe - \Delta X^{m-1} \Delta Z^{m-1} e \\ \mu e - SWe - \Delta S^{m-1} \Delta W^{m-1} \epsilon \end{array}\right]$$

the last equality being true by the induction hypothesis.

Thus, $\sum_{i=0}^{m} \Delta t^i$ and Δt^m are solution to the same system, which proves the proposition.

Proof of Theorem 3. The proof follows directly from Proposition 5 applied when $p = m_k$.

In order to complete the algorithm MPCLC we must specify the number of centering corrections at step 5. This number is dynamically selected, based on the complementarity that would result if a step were considered. Let g^i be the complementarity that would result if taking a step in the direction obtained after i corrections. We perform the $(i+1)$-th correction only if $g^i < g^{i-1}$ and i is less than the specified maximum number of corrections. If $g^i \geq g^{i-1}$ then the correcting process is interrupted and the direction Δt^{i-1} is used. This is the strategy considered in an experimental implementation of the MPCLC algorithm .

2.3 Cubic Convergence

The convergence analysis of the predictor-corrector algorithms follows directly from the books written by Ortega and Rheinboldt [1970] and by Dennis and Schnabel [1983], and the paper of Tapia, Zhang, Saltzman and Weiser [1996].

The pure Newton method applied to the system $F(t) = 0$, where $F(t)$ is given in (2.18) can be written as:

$$N(t) = t - F'(t)^{-1} F(t).$$

On the other hand, the predictor-corrector interior -point method can be written as:

$$\hat{N}(t) = t - \alpha F'(t)^{-1}[F(t) + F(N(t)) - \mu \hat{e}].$$

Therefore

$$\begin{gathered}\hat{N}(t) - t^* = t - t^* - F^{'}(t)^{-1}[F(t) + F(N(t))] + \\ (1-\alpha)F^{'}(t)^{-1}[F(t) + F(N(t))] + \alpha\mu F^{'}(t)^{-1}\hat{e} = \\ F^{'}(t)^{-1}[F(t)(N(t)) - F(N(t)) - F^{'}(t)t^*] + \\ (1-\alpha)F^{'}(t)^{-1}[F(t) + F(N(t))] + \alpha\mu F^{'}(t)^{-1}\hat{e} = \\ F^{'}(t)^{-1}\{F(N(t)) - F(t^*) - F^{'}(t^*)(N(t) - t^*)] + \\ [F^{'}(t^*) - F^{'}(t)](N(t) - t^*)\} + \\ (1-\alpha)F^{'}(t)^{-1}[F(t) + F(N(t))] + \alpha\mu F^{'}(t)^{-1}\hat{e}\ .\end{gathered}$$

In a neighbourhood of the solution t^* in the standard Newton method analysis we know that $\| N(t) - t^* \| = O(\| t - t^* \|^2)$.

But $\| F(t) \| = O(\| t - t^* \|)$ and $\| F(N(t)) \| = O(\| t - t^* \|^2)$. Hence, from the above relation we immediately obtain :

$$\begin{gathered}\| \hat{N}(t) - t^* \| = O(\| t - t^* \|^4) + O(\| t - t^* \|^3) + \\ | 1 - \alpha | O(\| t - t^* \|) + \mu O(1),\end{gathered}$$

which can be simplified to:

$$\begin{gathered}\| \hat{N}(t) - t^* \| = O(\| t - t^* \|^3) + \\ | 1 - \alpha | O(\| t - t^* \|) + \mu O(1). \qquad (2.29)\end{gathered}$$

Now, the term $\mu O(1)$ can be made $O(\| t - t^* \|^3)$ by the choice of μ. The only term which must be very carefully analysed is $| 1 - \alpha | O(\| t - t^* \|)$. It is quite clear that for cubic convergence we need $| 1 - \alpha |$ to be $O(\| t - t^* \|^2)$.

In the case of linear programming problems, for primal-dual interior -point method, Zhang, Tapia and Dennis [1992] obtained, assuming that strict complementarity, t_k is feasible, and t^* is a nondegenerate vertex solution, the following useful expression:

$$| 1 - \alpha_k | = \tfrac{1 - \tau_k + \sigma_k \theta_k}{1 - \sigma_k \theta_k} + O(x_k^T y_k), \quad (2.30)$$

where τ_k and σ_k are as in the PCLC algorithm, and $\theta_k \in (\frac{1}{n}, 1]$. This is also valid for the predictor-corrector interior -point method. Since for feasible t_k we have $x_k^T y_k = \| F(t_k) \|_1$, then $O(x_k^T y_k) = O(\| t - t^* \|)$.

The predictor-corrector methods could be viewed as a perturbed Newton method. The perturbation to the right-hand side is $\mu \hat{e} - F(t - F'(t)^{-1}F(t))$. From the above estimation of $| 1 - \alpha_k |$ we see that irrespective of the choices for τ_k and σ_k, this term is at best $O(\| t - t^* \|)$ and the predictor-corrector interior-point method cannot, even for nondegenerate problems, be shown to be cubically convergent. However, by choosing $\alpha_k = 1$ near solution and $\mu_k = O((x_k^T y_k)^3)$, from (2.29) we see that it is possible cubic convergence that is obtained. All these observations could be assembled into

Theorem 4. **Let $t_k = [x_k \ s_k \ y_k \ z_k \ w_k]^T$ be generated by the PCLC predictor- corrector interior- point algorithm with t_0 strictly feasible. Assume:**

1) Strict complementarity.

2) x^* is a nondegenerate vertex, and

3) The sequence $\{t_k\}$ converges to t^*.

If the choice of σ_k and τ_k satisfies:

$$0 \le \sigma_k \le \min(\sigma, c_1(x_k^T y_k)) \qquad (2.31)$$

and

$$0 \le \tau_k \le \min(\tau, 1 - c_2(x_k^T y_k)) \qquad (2.32)$$

where $\sigma \in [0,1), \tau \in (0.1)$ and $c_1, c_2 > 0$, then the convergence is quadratic, i.e. there is the positive constant $\gamma_1 > 0$ such that for k large

$$\| t_{k+1} - t^* \| \le \gamma_1 \| t_k - t^* \|^2 \ .$$

On the other hand, if instead of (2.31) we have

$$0 \le \sigma_k \le \min(\sigma, c_1(x_k^T y_k)^2) \qquad (2.33)$$

and for large k $\alpha_k = 1$, then the convergence is cubic, i.e. there exists the positive constant $\gamma_2 > 0$ such that for k large

$$\| t_{k+1} - t^* \| \le \gamma_1 \| t_k - t^* \|^3 \ .$$

Proof. The proof follows directly from the above discussions combined with some details given in [Zhang, Tapia and Dennis, 1992] and in [Tapia, Zhang, Saltzman and Weiser, 1996].

The remarkable aspect of Theorem 4 is that if into a neighbourhood of the solution the step-length one is selected (like in the Newton method), and the centering barrier parameter μ is of the order of the duality gap cubed (as Mehrotra suggests), then for nondegenerate problems, the predictor-corrector algorithms have a cubic convergence. Therefore, the predictor-corrector interior- point method and its variants for solving linear constrained optimization problems should be implemented in such a manner that near solution the centering barrier parameter is zero and the step-length is one, exactly as in the classical Newton method.

In order to consolidate the understanding of the behaviour of the predictor-corrector interior-point algorithms, we shall present a large-scale linear constrained optimization problem.

Example 3. Let us consider the problem:

$$\min \sum_{k=1}^{3m} x_k e^{-x_k}$$

subject to:

$$\begin{gathered} x_1 + 2x_m + x_{2m} = 5, \\ x_j + 2x_{m+j} + x_{2m+j} = 10 + \frac{1}{j}, \\ (j = 1, ..., m), \\ 0 \leq x_j \leq 10. \end{gathered}$$

The PCLC and MPCLC algorithms, initialized in the point: $x_0 = [0.1...0.1]^T$, for different values of m, give the results from the following Table.

m	n	vfo
500	1500	84.279940
1000	3000	168.469280
1200	3600	202.149807
1500	4500	252.673138
2000	6000	336.882943
3000	9000	505.311133

PCLC			MPCLC		
it	nf	Time	it	nf	Time
10	11	0:0:15:77	8	9	0:0:10:44
10	11	0:0:57:62	8	9	0:0:32:40
10	11	0:1:19:91	8	9	0:0:43:50
10	11	0:2:0:67	8	9	0:1:4:3
10	11	0:3:40:30	8	9	0:1:55:02
10	11	0:7:41:54	8	9	0:3:52:50

m = number of constraints
n = number of variables
vfo = value of the objective function
it = number of iterations
nf = number of objective function evaluations

2.4 Alternative Formulation

The systems of Eqs (2.8) and (2.14) with the coefficient matrix

$$\begin{bmatrix} A & & & & \\ I & I & & & \\ -\nabla^2 f(x) & & A^T & I & -I \\ Z & & & X & \\ & W & & & S \end{bmatrix} \qquad (2.34)$$

are usually not solved directly, instead they are algebraically simplified.

Firstly, simple algebraic manipulations show that the solving of the above linear systems could be reduced to that of factorizing the matrix:

$$\begin{bmatrix} -[\nabla^2 f(x) + X^{-1}Z + S^{-1}W] & A^T \\ A & \end{bmatrix} \qquad (2.35)$$

Note that the matrix (2.35) is symmetric whereas the original matrix (2.34) is not. On the other hand, the reduction process from (2.34) to (2.35) entails no off-diagonal fill-in in the system of equations, which is very appealing. However, the matrix (2.35) is indefinite.

The factorization of symmetric indefinite matrices, like (2.35) has been considered by Vanderbei and Carpenter [1993] (the LoQo and ALPO packages for linear programming, [Vanderbei, 1990a, 1990b, 1991]), Gill, Saunders and Shinnerl [1996], Gill, Murray, Ponceleón and Saunders [1995], Fourer and Mehrotra [1991]. Gill, Murray, Ponceleón and Saunders [1995], and Fourer and Mehrotra [1991] show that performing the Bunch-Parlett [1971] factorization of the indefinite matrix (2.35) will prevent fill-in.

Secondly, it is quite clear that the reduction process of the above linear systems (2.8) and (2.14) could be continued, finally the computation of the predictor-corrector direction involving the factorization of the symmetric matrix:

$$A(\nabla^2 f(x) + X^{-1}Z + S^{-1}W)^{-1}A^T.(2.36)$$

Due to the convexity of the function $f(x)$, the advantage of factorizing the matrix (2.36) over (2.35) is in that it is based on a symmetric positive definite matrix guaranteed to yield a factorization of the form $L\Lambda L^T$, where L is a lower triangular matrix with unit diagonal elements, and Λ is a diagonal matrix with strictly positive entries.

However, the great disadvantage of solving the system with matrix (2.36) is that it suffers a great deal of fill-in when (2.36) is formed. This is especially true due to the presence of the Hessian $\nabla^2 f(x)$ of the objective function, and if the A matrix has a number of dense columns.

For linear programming problems, most of the implementations of interior- point methods, like those of Lustig, Marsten and Shanno [1991, 1992, 1994] (the OB1 package), Carpenter, Lustig, Mulvey and Shanno [1993] (the OBN package), Gondzio [1996] (the HOPDM package), Andersen and Andersen [1997] (the APOS package), or Meszaros [1994] (the BPMPD package) are based on factorizing the matrix (2.36) where $\nabla^2 f(x) = 0$.

For linear constrained optimization problems, in an experimental code, we shall consider the solving of the linear system with the coefficient matrix (2.35).

3 Numerical Examples

In this Section we shall present some numerical examples.

Problem P1. [Hock and Schittkowski, 1981], Problem 112, page 121, Chemical equilibrium)

$$\min \sum_{j=1}^{10} x_j (c_j + \ln \frac{x_j}{x_1+\ldots+x_{10}})$$

subject to:

$$x_1 + 2x_2 + 2x_3 + x_6 + x_{10} = 2,$$
$$x_4 + 2x_5 + x_6 + x_7 = 1,$$
$$x_3 + x_7 + x_8 + 2x_9 + x_{10} = 1.$$

Table P1 ($\sigma = 0.1, \tau_k = 0.99995, \varepsilon = 10^{-8}$)

c_j	*Lower* *bound*	x_0	x^*	*Upper* *bound*
-6.089	0.0001	0.1	0.040668	100.0
-17.164	0.0001	0.1	0.147732	100.0
-34.054	0.0001	0.1	0.783150	100.0
-5.914	0.0001	0.1	0.001414	100.0
-24.721	0.0001	0.1	0.485246	100.0
-14.986	0.0001	0.1	0.000693	100.0
-24.100	0.0001	0.1	0.27399	100.0
-10.708	0.0001	0.1	0.017947	100.0
-26.662	0.0001	0.1	0.037314	100.0
-22.179	0.0001	0.1	0.096872	100.0

$f(x^*) = -47.76109$.

Problem P2.

$$\min \sum_{k=1}^{13} (\frac{1}{k} x_k^2 + k x_k + k^2)^2$$

subject to:

$$x_1 + x_3 + x_5 + x_7 + x_9 = 10,$$
$$x_2 + 2x_4 + 3x_6 + 4x_8 + 5x_{10} = 20,$$
$$2x_2 - 5x_5 + 8x_4 = 30,$$
$$-x_1 + 3x_4 - 5x_7 + x_{10} - x_{11} = 0,$$
$$x_1 + 2x_2 + 4x_4 + 8x_8 + x_{12} = 100,$$
$$x_1 + 3x_3 + 6x_6 - 9x_9 + x_{13} = 50.$$

Table P2. ($\sigma = 0.1, \tau_k = 0.99995, \varepsilon = 10^{-8}$)

Lower *bound*	x_0	x^*	*Upper* *bound*
0.0	0.1	5.666234	100.0
0.0	0.1	4.966837	100.0
0.	0.1	4.333765	100.0
0.0	0.1	2.5	100.0
0.0	0.1	0.0	100.0
0.0	0.1	0.0	100.0
0.0	0.1	0.0	100.0
0.0	0.1	2.5	100.0
0.0	0.1	0.0	100.0
0.0	0.1	0.0	100.0
0.0	0.1	1.833765	100.0
0.0	0.1	54.33377	100.0
0.0	0.1	31.33247	100.0

$f(x^*) = 1562547.3$

Problem P3.

$$\min \sum_{k=1}^{15} x_k^2 + \sum_{k=1}^{14} (x_k + x_{k+1})$$

subject to:

$$0.1x_1 + 0.1x_7 + 0.3x_8 + 0.2x_9 + 0.2x_{11} = 1,$$
$$0.1x_2 + 0.2x_8 + 0.3x_9 + 0.4x_{10} + x_{11} = 2,$$
$$0.1x_3 + 0.2x_8 + 0.3x_9 + 0.3x_9 + 0.4x_{10} + 2x_{11} = 3,$$
$$x_4 + x_8 + 0.5x_9 + 0.5x_{10} + x_{11} - x_{12} = 3,$$
$$2x_5 + x_6 + 0.5x_7 + 0.5x_8 + 0.23x_9 + 0.25x_{10} + 0.5x_{11} - x_{13} = 4,$$
$$x_4 + x_6 + x_8 + x_9 + x_{10} + x_{11} - x_{14} = 5,$$
$$0.1x_1 + 1.2x_7 + 1.2x_8 + 1.4x_9 + 1.1x_{10} + 2x_{11} - x_{15} = 6.$$

Table P3. ($\sigma = 0.1, \tau_k = 0.99995, \varepsilon = 10^{-8}$)

Lower *bound*	x_0	x^*	*Upper* *bound*
0.0	0.1	0.452456	100.0
0.0	0.1	0.266273	100.0
0.0	0.1	0.0	100.0
0.0	0.1	0.0	100.0
0.0	0.1	1.096372	100.0
0.0	0.1	0.048186	100.0
0.0	0.1	0.0	100.0
0.0	0.1	1.492774	100.0
0.0	0.1	1.507982	100.0
0.0	0.1	0.489488	100.0
0.0	0.1	1.026627	100.0
0.0	0.1	0.518137	100.0
0.0	0.1	0.0	100.0
0.0	0.1	0.0	100.0
0.0	0.1	0.539443	100.0

$f(x^*) = 21.718976$.

Problem P4.

$$\min[3x_1 e^{-0.1 x_1 x_6} + 4x_2 + x_3^2 + 7x_4 + \frac{10}{x_5} + x_6]$$

subject to:

$$-x_4 + x_5 - x_6 = 0.1,$$
$$x_1+x_2+x_3+x_4+x_5+x_6+x_7+x_8+x_9+x_{10} = 20,$$
$$8x_1 + 3x_2 + 4x_3 + x_4 + x_5 + x_7 = 10,$$
$$2x_1 + 6x_2 + x_3 + 3x_4 + x_6 + x_8 = 13,$$
$$2x_1 + x_2 + x_3 + 3x_4 - x_9 = 2,$$
$$x_1 + 4x_2 + 5x_3 + 2x_4 + x_{10} = 18.$$

Table P4. ($\sigma = 0.1, \tau_k = 0.99995, \varepsilon = 10^{-8}$)

Lower *bound*	x_0	x^*	*Upper* *bound*
0.0	0.1	0.0	10.0
0.0	0.1	0.893288	10.0
0.0	0.1	1.146728	10.0
0.0	0.1	0.0	10.0
0.0	0.1	2.733221	10.0
0.0	0.1	2.633221	10.0
0.0	0.1	0.0	10.0
0.0	0.1	3.860319	10.0
0.0	0.1	0.040016	10.0
0.0	0.1	8.693204	10.0

$f(x^*) = 11.180048.$

Problem P5.

$$\min \sum_{k=1}^{5} (x_k - x_{k+1})^2 + \sum_{k=6}^{10} (x_k - x_{k+1})^4$$

subject to:

$$x_k + 2x_{k+1} + 3x_{k+2} = 6,$$
$$(k = 1, ..., 8)$$
$$x_9 + 2x_{10} + 3x_{11} = 9.$$

Table P5. ($\sigma = 0.1, \tau_k = 0.99995, \varepsilon = 10^{-8}$)

Lower *bound*	x_0	x^*	*Upper* *bound*
0.0	0.1	1.016505	10.0
0.0	0.1	1.004545	10.0
0.0	0.1	0.991468	10.0
0.0	0.1	1.004173	10.0
0.0	0.1	1.000062	10.0
0.0	0.1	0.998567	10.0
0.0	0.1	1.000934	10.0
0.0	0.1	0.999854	10.0
0.0	0.1	0.999785	10.0
0.0	0.1	1.000191	10.0
0.0	0.1	1.999944	10.0

$f(x^*) = 0.9995046.$

Problem P6.

$$\min \sum_{k=1}^{m+2} x_k^2$$

subject to:

$$x_1 + 2x_2 + x_3 = 3,$$
$$x_1 + x_j + 2x_{j+1} + x_{j+2} = 4,$$
$$(j = 2, ..., m).$$

Table P6 (m=20)
($\sigma = 0.1, \tau_k = 0.99995, \varepsilon = 10^{-8}$)

Lower *bound*	x_0	x^*	*Upper* *bound*
0.0	0.1	2.003768	10.0
0.0	0.1	0.031407	10.0
0.0	0.1	0.933417	10.0
0.0	0.1	0.097989	10.0
0.0	0.1	0.866834	10.0
0.0	0.1	0.164572	10.0
0.0	0.1	0.800251	10.0
0.0	0.1	0.231155	10.0
0.0	0.1	0.733668	10.0
0.0	0.1	0.297738	10.0
0.0	0.1	0.667085	10.0
0.0	0.1	0.364321	10.0
0.0	0.1	0.600502	10.0
0.0	0.1	0.430904	10.0
0.0	0.1	0.533919	10.0
0.0	0.1	0.497487	10.0
0.0	0.1	0.467336	10.0
0.0	0.1	0.564070	10.0
0.0	0.1	0.400753	10.0
0.0	0.1	0.630653	10.0
0.0	0.1	0.334170	10.0
0.0	0.1	0.697236	10.0

$f(x^*) = 10.345477.$

Problem P7.

$$\min \sum_{k=1}^{m+4} (x_k^2 + 2x_k + 6)e^{-x_k}$$

subject to:

$$x_1 + 2x_3 + x_5 = 4,$$
$$x_j + 2x_{j+2} + x_{j+4} = 8,$$
$$(j = 2, ..., m)$$

Table P7 (m=10)
($\sigma = 0.1, \tau_k = 0.99995, \varepsilon = 10^{-8}$)

Lower *bound*	x_0	x^*	*Upper* *bound*
0.0	0.1	0.274447	10.0
0.0	0.1	2.286694	10.0
0.0	0.1	0.202727	10.0
0.0	0.1	1.713305	10.0
0.0	0.1	3.320098	10.0
0.0	0.1	2.286695	10.0
0.0	0.1	1.157077	10.0
0.0	0.1	1.713305	10.0
0.0	0.1	2.365749	10.0
0.0	0.1	2.286695	10.0
0.0	0.1	2.111426	10.0
0.0	0.1	1.713305	10.0
0.0	0.1	1.411399	10.0
0.0	0.1	2.286695	10.0

$f(x^*) = 33.242885.$

Problem P8.

$$\min \sum_{k=1}^{3m} \frac{x_k}{x_k+1} \ln x_k$$

subject to:

$$x_1 + 2x_{m+1} + x_{2m+1} = 4,$$
$$x_j + 2x_{m+j} + x_{2m+j} = 8 + \frac{1}{j},$$
$$(j = 2, \dots, m)$$

Table P8 (m=5)
$(\sigma = 0.1, \tau_k = 0.99995, \varepsilon = 10^{-8})$

Lower *bound*	x_0	x^*	*Upper* *bound*
0.0	0.1	0.386228	10.0
0.0	0.1	0.332300	10.0
0.0	0.1	0.333421	10.0
0.0	0.1	0.333997	10.0
0.0	0.1	0.334348	10.0
0.0	0.1	1.613771	10.0
0.0	0.1	3.917699	10.0
0.0	0.1	3.833245	10.0
0.0	0.1	3.791002	10.0
0.0	0.1	3.765651	10.0
0.0	0.1	0.386228	10.0
0.0	0.1	0.332300	10.0
0.0	0.1	0.333421	10.0
0.0	0.1	0.333997	10.0
0.0	0.1	0.334348	10.0

$f(x^*) = 1.824042.$

In oder to compare the interior -point methods with the classical methods we shall solve these problems with the following packages for mathematical programming:

Name Package	Author
TOLMIN	M.J.D. Powell University of Cambridge
PCLC	N.Andrei ICI - Bucharest
NLPQL	K.Schittkowski Bayreuth University
MINOS	B.Murtagh, M.Saunders Stanford University
SPENBAR	N.Andrei ICI Bucharest

The characteristics of the solving process corresponding to the P1-P8 linear optimization problems are presented in the following Tables.

We emphasize that the TOLMIN and PCLC packages are specialized in linear optimization problems. The MINOS package is for large-scale nonlinear optimization and NLPQL and SPENBAR are dedicated to solving small- or medium-scale, strongly nonlinear programming problems

Prob	n	m
P1	10	3
P2	13	6
P3	15	7
P4	10	6
P5	11	9
P6	22	20
P7	14	10
P8	15	5

TOLMIN			PCLC		
it	nf	Time	it	nf	Time
28	36	0:0:0:16	8	9	0:0:0:83
22	34	0:0:0:16	37	38	0:0:3:40
9	13	0:0:0:6	9	10	0:0:1:15
5	16	0:0:0:5	38	39	0:0:2:80
4	8	0:0:0:5	8	9	0:0:1:05
7	13	0:0:0:16	9	10	0:0:0:15
12	33	0:0:0:16	12	13	0:0:0:17
13	20	0:0:0:17	9	10	0:0:0:11

Prob	NLPQL			MINOS				SPENBAR			
	it	nf	Time	oit	init	nf	Time	oit	init	nf	Time
P1	21	30	0:0:0:44	1	19	46	0:0:0:27	6	228	1263	0:0:5:77
P2	7	1045	0:0:0:50	1	18	15	0:0:0:17	9	395	2308	0:0:14:94
P3	5	5	0:0:0:16	1	18	14	0:0:0:22	9	276	1898	0:0:14:44
P4	10	10	0:0:0:17	1	12	8	0:0:0:11	9	401	2384	0:0:12:74
P5	6	7	0:0:0:39	1	12	9	0:0:0:16	6	94	515	0:0:3:41
P6	5	7	0:0:2:09	1	25	11	0:0:0:38	8	1021	10877	0:3:13:95
P7	16	16	0:0:1:59	1	20	21	0:0:0:27	5	156	991	0:0:8:73
P8	14	18	0:0:1:10	1	15	26	0:0:0:16	4	54	303	0:0:2:41

n = number of variables
m = number of constraints
it = number of iterations
init = number of inner (minor) iterations
oit = number of outer (major) iterations
nf = number of objective function evaluations

In the following we shall present a comparison among these optimization packages corresponding to the above P6-P8 problems in case of medium- and large- scale dimensions.

The following Tables give the characteristics of solving the problem **P6**, for different values of m, with TOLMIN and PCLC packages and MINOS and NLPQL packages, respectively.

m	n	vfo	TOLMIN			PCLC		
			it	nf	Time	it	nf	Time
20	22	10.345477	7	13	0:0:0:22	9	10	0:0:0:15
40	42	16.464030	3	7	0:0:0:72	10	11	0:0:0:59
60	62	22.664322	4	20	0:0:1:64	10	11	0:0:1:15
80	82	28.889140	3	8	0:0:3:18	9	10	0:0:1:92
100	102	35.123936	2	6	0:0:5:83	11	12	0:0:4:22
200	202	66.343333	6	30	0:0:46:47	6	7	0:0:5:76

m	n	vfo	MINOS			NLPQL		
			it	nf	Time	it	nf	Time
20	22	10.345477	25	11	0:0:0:39	5	7	0:0:2:03
40	42	16.464030	43	7	0:0:0:77	3	4	0:0:6:92
60	62	22.664322	63	7	0:0:1:32	3	4	0:0:19:61
80	82	28.889140	85	7	0:0:2:15	3	4	0:0:41:03
100	102	35.123936	104	6	0:0:3:02	4	4	0:1:46:28
200	202	66.343333	204	6	0:0:9:94	2	2	0:5:55:53

For large scale dimensions the MINOS and PCLC packages have the following behaviour:

m	n	vfo	MINOS			PCLC		
			it	nf	Time	it	nf	Time
500	502	160.074858	503	6	0:0:50:20	11	12	0:1:12:19
1000	1002	316.318683	1003	6	0:2:42:36	11	12	0:4:38:36
1200	1202	378.817653	1203	6	0:3:51:29	11	12	0:5:37:47
1500	1502	472.566623	a			11	12	0:9:16:45
2000	2002	628.815592	a			11	12	0:16:13:12
3000	3002	941.314562	a			11	12	1:4:24:6

a = insufficient memory

For the problem **P7** the following Tables illustrate the behaviour of TOLMIN and PCLC packages and of MINOS and NLPQL, respectively, for different values of m.

m	n	vfo	TOLMIN			PCLC		
			it	nf	Time	it	nf	Time
20	24	52.688309	11	30	0:0:0:38	11	12	0:0:0:28
40	44	93.878536	11	25	0:0:1:26	10	11	0:0:0:55
60	64	134.915594	5	7	0:0:2:26	10	11	0:0:1:05
80	84	175.917332	7	25	0:0:4:51	10	11	0:0:1:82
100	104	216.893341	7	24	0:0:7:31	10	11	0:0:2:58
200	204	421.719855	11	43	0:0:49:10	10	11	0:0:9:51

m	n	vfo	MINOS			NLPQL		
			it	nf	Time	it	nf	Time
20	24	52.688309	16	28	0:0:0:44	16	16	0:0:6:70
40	44	93.878536	46	13	0:0:0:83	10	10	0:0:22:41
60	64	134.915594	66	12	0:0:1:43	10	10	0:1:5:04
80	84	175.912732	86	12	0:0:2:19	10	10	0:2:22:09
100	104	216.893341	106	12	0:0:3:7	10	10	0:4:25:51
200	204	421.719855	206	13	0:0:10:5	12	12	0:39:40:42

For large -scale dimensions the MINOS and PCLC give the results from the Table below:

m	n	vfo	MINOS			PCLC		
			it	nf	Time	it	nf	Time
500	504	1036.073610	509	19	0:0:49:15	15	16	0:1:37:22
1000	1004	2059.953591	1010	20	0:2:40:05	11	12	0:5:3:62
1200	1204	2469.503050	1210	20	0:3:35:96	12	13	0:8:19:77
1500	1504	3083.826330	a			13	14	0:15:11:49
2000	2004	4107.697251	a			13	14	0:25:26:21
3000	3004	6155.437271	a			14	15	1:04:48:56

a = insufficient memory.

The Tables which follow give the results of solving the problem **P8** with TOLMIN and PCLC and MINOS and NLPQL, respectively, for different values of m.

m	n	vfo	TOLMIN			PCLC		
			it	nf	Time	it	nf	Time
20	60	9.076395	38	74	0:0:4:78	9	10	0:0:0:17
40	120	18.600455	41	81	0:0:22:85	9	10	0:0:0:39
60	180	28.086333	36	81	0:0:54:49	10	11	0:0:0:71
80	240	37.556382	37	95	0:1:52:44	10	11	0:0:0:99
100	300	47.017712	37	106	0:3:20:59	10	11	0:0:1:38
200	600	94.267018	36	106	0:21:49:59	11	12	0:0:4:45

m	n	vfo	MINOS			NLPQL		
			it	nf	Time	it	nf	Time
20	60	9.076395	33	31	0:0:0:99	17	20	0:0:45:81
40	120	18.600455	58	42	0:0:3:42	20	21	0:6:5:86
60	180	28.086333	78	42	0:0:6:43	20	22	0:19:42:98
80	240	37.556382	98	42	0:0:10:55	18	19	0:41:55:54
100	300	47.017712	118	41	0:0:15:82	20	22	1:30:24:22
200	600	94.267018	415	631	0:4:58:8	b		

b = Execution time greater that 4 hours

For large -scale dimensions the PCLC package gives the following results:

m	n	vfo	PCLC		
			it	nf	Time
500	1500	235.856441	12	13	0:0:22:8
1000	3000	471.724819	12	13	0:1:21:2
1200	3600	566.059177	12	13	0:1:57:10
1500	4500	707.553823	12	13	0:3:0:54
2000	6000	943.366696	12	13	0:5:10:44
3000	9000	1414.969173	12	13	0:11:21:79

It is very instructive to notice the behaviour of the MPCLC algorithm in case of the problem P8 in which the maximum number of centering corrections at each iteration is limited to 5. The Table below illustrates this behaviour for different values of m.

m	n	vfo	MPCLC		
			it	nf	Time
500	1500	235.856441	8	9	0:0:15:38
1000	3000	471.724819	8	9	0:0:52:78
1200	3600	566.059177	8	9	0:1:15:47
1500	4500	707.553823	8	9	0:1:51:33
2000	6000	943.366696	8	9	0:3:12:30
3000	9000	1414.969173	8	9	0:6:58:42
4000	12000	1886.555515	8	9	0:12:21:22

The analysis of the results obtained on this limited set of problems illustrates that the predictor-corrector interior- point algorithms are a powerful alternative to solving linear constrained optimization problems.

Still, PCLC (and MPCLC) is a development code, but we notice that for small- scale problems all the packages have the same behaviour. Significant differences appear in the case of medium- and large -scale problems.

For predictor-corrector interior- point algorithms the number of iterations and function evaluations seems to be independent of the number of variables or constraints.

For the class of problems P6-P8, with a number of constraints between 20 and 200, from the above Tables it follows that the total time (in milliseconds) needed by the 4 packages considered here is given in the Table below:

Problem	Packages			
	TOLMIN	PCLC	MINOS	NLPQL
P6	5806	1379	1759	53140
P7	6482	1579	1801	291217
P8	170474	809	33529	

It follows that for medium- scale linear constrained optimization problems the PCLC and MINOS are the most suited packages. Anyway, MINOS remains the state-of-the-art and the most respectable package for large-scale optimization. For the moment, TOLMIN does not implement sparse matrices techniques. Introducing these techniques in TOLMIN could result in a very competitive package.

4 Conclusion

In this paper we have extended the predictor-corrector interior -point approach to the linear constraint optimization problems with convex, nonlinear objective function. The idea used in linear programming case and implemented in some state-of-the-art predictor-corrector interior- point packages is also considered in the case of linear constrained problems. Firstly, the algorithms allow iterates to be very close to the boundary of the positive orthant, and secondly they correct this step using a centering step. The theorems proved here provide a theoretical justification for such a practice in fast convergence.

The numerical results presented here are obtained with a crude experimental code which is still in progress of development. We expect improvements both in terms of efficiency and of robustness. These could mainly be obtained using advanced strategies for solving augmented indefinite linear systems like (2.35). The very good stability properties of the augmented system approach motivated their incorporation into several interior- point codes [Duff, Gould, Reid, Scott and Turner, 1991], [Fourer and Mehrotra, 1993], [Maros and Meszaros, 1995], Turner [1991] and [Vanderbei and Carpenter, 1993]. Other advantages of this approach, namely an easy to do handling of dense columns in A and the ability of its easy extension to the quadratic programming problems emphasized the importance of the augmented system approach.

It is quite clear that the success of the augmented system factorization highly depends on the efficiency of the pivot selection strategy. For the class of problems considered here this strategy must also take into consideration the structure of the Hessian matrix of the objective function. This will introduce some complications. We must find a "good" partition both for A and $\nabla^2 f(x)$ matrices in order to determine which block can be inexpensively pivoted out and which one should be delayed as much as possible. These algebraic mechanisms are under current consideration.

Acknowledgments

I gratefully acknowledge to the Alexander von Humboldt Foundation for their constant moral, financial and material support generously offered for more than 2 years, which I spent at Duisburg and Bayreuth Universities - Germany.

Special thanks are due to Professor Klaus Schittkowski (Bayreuth University) for his stimulating discussions and availability during my stay in Bayreuth, to Professor Michael Saunders (Stanford University-USA) for giving me the MINOS code, and some other practical hints at its usage, to Professor M.J.D. Powell for TOLMIN package and some other papers, to Dr. Marc Breitfeld and Professor David Shanno (Rutgers University) for 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 , Denmark, 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., PCIP - **Predictor Corrector Interior Point Algorithm for Large-scale Linear Programming.** Technical Paper No. AMOL-97-1, Research Institute for Informatics, Bucharest, March 1997.

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 1997.

ANDREI, N., **Programarea Matematica Avansata. Teorie, Metode computationale, Aplicatii (in Romanian), a** manuscript submitted to the Publishing House of the Romanian Academy, 1997.

ANSTREICHER, K.M., **On Long Step Path Following and SUMT for Linear and Quadratic Programming.** Manuscript, Department of Operations Research, Yale University, New Haven, 1990.

BARNES, E.R., **A Variation on Karmarkar's Algorithm for Solving Linear Programming Problems**, MATHEMATICAL PROGRAMMING, Vol.36, 1986, pp.174-182.

BAYER, D.A. and LAGARIAS, J.C., **The Nonlinear Geometry of Linear Programming, Part I: Affine and Projective Scaling Trajectories**, Trans. Amer. Math. Soc., Vol.314, 1989, pp.499-526. **Part II: Legendre Transform Coordinates**, Trans. Amer. Math. Soc., Vol.314, 1989, pp.527-581.

BEIGLER, L.T., NOCEDAL, J., SCHMID, C.and TERNET, D., **Numerical Experience With A Reduced Hessian Method for Large Scale Constrained Optimization**, Technical Report OTC 97/06, Argonne National Laboratory, 1997.

BUNCH, J.R. and PARLETT, B.N., **Direct Methods for Solving Symmetric Indefinite Systems of Linear Equations**, SIAM JOURNAL ON NUMERICAL ANALYSIS, Vol.8, 1971, pp.639-655.

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, 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.

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

DIKIN, I.I., **Iterative Solution of Problems of Linear and Quadratic Programming**. Soviet Mathematical Reports, Vol.8, 1967, pp.674-675.

DOMICH, P.D., BOGGS, P.T., DONALDSON, J.R. and WITZGALL, C., **Optimal 3-Dimensional Methods for Linear Programming**. NISTIR 89-4225, Center for Computing and Applied Mathematics, U.S. Department of Commerce, National Institute of Standards and Technology, 1989.

DUFF, I.DS., GOULD, N.I.M., REID, J.K., SCOTT, J.A. and TURNER, K. **The Factorization of Sparse Symmetric Indefinite Matrices**, IMA JOURNAL ON NUMERICAL ANALYSIS, Vol.11, 1991, pp.181-204.

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 APPLICATION, Vol.89, No.3, 1996, pp.507-541.

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

FOURER, R. and MEHROTRA, S., **Performance of An Augmented System Approach for Solving Least-squares Problems in An Interior Point Method for Linear Programming**. Mathematical Programming Society COAL NEWSLETTER, Vol.19, 1991, pp.26-31.

FOURER, R. and MEHROTRA, S., **Solving Symmetric Indefinite Systems in An Interior-point Method for Linear Programming**, MATHEMATICAL PROGRAMMING, Vol.62, 1993, pp.15-39.

FREUND, R.M., **Theoretical Efficiency of A Shifted Barrier Function Algorithm for Linear Programming**, LINEAR ALGEBRA AND ITS APPLICATIONS, Vol.152, 1991, pp.19-41.

FRISCH, K.R., **The Logarithmic Potential Method for Convex Programming**, Manuscript, Institute of Economics, University of Oslo, Oslo, May 1955.

GAY, D.M. **Electronic Mail Distribution of Linear Programming Test Problems**. Mathematical Programming Society COAL NEWSLETTER, 1988.

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., SAUNDERS, M.A. and SHINNERL, J.R., **On the Stability of Cholesky Factorization for Symmetric Quasidefinite Systems**, SIAM JOURNAL ON MATRIX ANALYSIS AND APPLICATIONS, Vol.17, No.1, January 1996, pp.35-46.

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., **Splitting Dense Columns of Constraint Matrix in Interior Point**

Methods for Large Scale Linear Programming, OPTIMIZATION, Vol.24, 1992, pp.285-297.

GONDZIO, J., **Implementing Cholesky Factorization for Interior Point Methods of Linear Programming**, OPTIMIZATION, Vol. 27, 1993, pp.121-140.

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.

GONZAGA, C.C., **An Algorithm for Solving Linear Programming Problems in $O(n^3L)$** *Operations*, in: N. Megiddo (Ed.) Progress in Mathematical Programming. Interior-Point and Related Methods, SPRINGER-VERLAG, New York, 1989, pp.1-28.

GONZAGA, C.C., **Polynomial Affine Algorithms for Linear Programming**, MATHEMATICAL PROGRAMMING, Vol.49, 1990, pp.7-21.

GONZAGA, C.C., **Interior Point Algorithms for Linear Programming Problems With Inequality Constraints.** MATHEMATICAL PROGRAMMING, Vol.52, 1991, pp.209-226.

GONZAGA, C.C., **Path-following Methods for Linear Programming**, SIAM REVIEW, Vol.34, 1992, pp.167-224.

GONZAGA, C.C. and TODD, M., **An $O(\sqrt{n}L)$-iteration Large-step Primal-dual Affine Algorithm for Linear Programming**, SIAM JOURNAL ON OPTIMIZATION, Vol.2, 1992.

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

JANSEN, B., ROSS, C. and TERLAKY, T., **Target-following Methods for Linear Programming**, in T. Terlaky (Ed.) Interior Point Methods of Mathematical Programming, KLUWER ACADEMIC PUBLISHERS, Dordrecht, 1996, pp.83-124.

KARMARKAR, N., **A New Polynomial-time Algorithm for Linear Programming**, COMBINATORICA, Vol.4, 1984, pp.373-395.

KARMARKAR, N., **A New Polynomial-time Algorithm for Linear Programming**, presented at the ORAS/TIMS Meeting, Boston, November 1985.

KHACHIYAN, L.G., **A Polynomial Algorithm in Linear Programming.** Soviet Mathematics Reports, Vol.20, No.1, 1979, pp.191-194. [Doklady Akademiia Nauk SSSR, Vol. 244, 1979, pp.1093-1096.]

KHACHIYAN, L.G., **Polynomial Algorithms in Linear Programming.** USSR Computational Mathematics and Mathematical Physics, Vol.20, 1980, pp.53-72. [Zhurnal Vychisditel'noi Matematiki i Matematicheskoi Fiziki, Vol.20, 1980, pp.51-68.]

KIM, K. and NAZARETH, J.L., **A Primal Null Space Affine Scaling Method**, ACM TRANSACTIONS ON MATHEMATICAL SOFTWARE, Vol.20, No.3, 1994, pp.373-392.

KOJIMA, M., MIZUNO, S. and YOSHISE, A., **A Polynomial-time Algorithm for A Class of Linear Complementarity Problems**, MATHEMATICAL PROGRAMMING, Vol.44, 1989, pp.1-26.

KOJIMA, M., MIZUNO, S. and YOSHISE, A., **A Primal-dual Interior Point Algorithm for Linear Programming**, in N. Megiddo (Ed.) Progress in Mathematical Programming. Interior-Point and Related Methods, SPRINGER-VERLAG, New York, 1989, pp.29-47.

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.

LUSTIG, I.J., **A Generic Primal-dual Interior Point Algorithm**, Technical Report SOR 88-3, Princeton University, 1988.

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.

MAROS, I. and MESZAROS, C., **The Role of the Augmented System in Interior**

Point Methods, Technical Report TR/06/95, Brunel University, Department of Mathematics and Statistics, London, 1995.

MEGIDDO, N., **Pathways To the Optimal Set in Linear Programming**, in N. Megiddo (Ed.) Progress in Mathematical Programming, Interior-point and Related Methods, SPRINGER-VERLAG, New York, 1989, pp.131-158.

MEHROTRA, S., **Variants of Karmarkar's Algorithm: Theoretical Complexity and Practical Implementation**, Ph.D Thesis, submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Graduate School of Arts and Science, 1987.

MEHROTRA, S., **On Finding A Vertex Solution Using Interior Point Methods**, LINEAR ALGEBRA AND ITS APPLICATIONS, Vol.152, 1990, pp.233-253.

MEHROTRA, S., **On the Implementation of A Primal-dual Interior Point Method**, SIAM JOURNAL ON OPTIMIZATION, Vol.2, No.4, November 1992, pp.575-601.

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

MIZUNO, S., TODD, M.J. and YE, Y., **On Adaptive Step Primal Dual Interior Point Algorithms for Linear Programming**, Technical Report No. 944, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York, 1990.

MONTEIRO, R.D.C. and ADLER, I., **Interior Path-following Primal-dual Algorithms. Part I: Linear Programming**, MATHEMATICAL PROGRAMMING, Vol.44, 1989, pp.27-41.

MONTEIRO, R.D.C. and ADLER, I., **Interior Path-following Primal-dual Algorithms. Part II: Convex Quadratic Programming**, MATHEMATICAL PROGRAMMING, Vol.44, 1989, pp.42-66.

MONTEIRO, R.C., ADLER, I. and RESENDE, M.G.C., **A Polynomial Time Primal Dual Affine Scaling Algorithm for Linear and Convex Quadratic Programming and Its Power Series Extension**, MATH. OPER. RES., Vol.15, 1990, pp.191-214.

NAZARETH, J.L., **Pricing Criteria in Linear Programming**, in N. Megiddo (Ed.) Progress in Mathematical Programming, Interior Point and Related Methods, SPRINGER-VERLAG, New York, 1989, pp.105-129.

ORTEGA, J.M. and RHEINBOLDT, W.C., **Iterative Solution of Nonlinear Equations in Several Variables**, ACADEMIC PRESS, New York, 1970.

POTRA, F.A., **An Infeasible Interior Point Predictor Corrector Algorithm for Linear Programming**, SIAM JOURNAL ON OPTIMIZATION, Vol.6, No.1, 1996, pp.19-32.

RENEGAR, J.A., **A Polynomial-time Algorithm Based On Newton's Method for Linear Programming**, MATHEMATICAL PROGRAMMING, Vol.40, 1988, pp.59-93.

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

SHOR, N.Z., **Convergence Rate of the Gradient Descent Method With Dilatation of the Space**, CYBERNETICS, Vol.6, No.2, 1970, pp.102-108.

SHOR, N.Z., **Utilization of the Operation of Space Dilatation in the Minimization of Convex Functions**, KIBERNETIKA, Vol.6, No.1, Kiev, 1970, pp.6-12. [Translation into English in: CYBERNETICS, Vol.6, No.1, 1970, pp.7-15.]

SHOR, N.Z., **Convergence Rate of the Gradient Descent Method With Dilatation of the Space**, KIBERNETIKA, Vol.6, No.2, Kiev, 1970, pp.80-85.

SONNEVEND, G., **An Analytical Centre for Polyhedrons and New Classes of Global Algorithms for Linear (Smooth, Convex) Programming.** Lecture Notes Control Information Sci., Vol.84, SPRINGER-VERLAG, New York, 1985, pp.866-876.

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.

TODD, M.J. and YE, Y., **A Centered Projective Algorithm for Linear Programming**, MATH. OPER. RES., Vol.15, 1990, pp.508-529.

TRAUB, J.F., **Iterative Methods for the Solution of Equations**, PRENTICE-HALL, Inc., Englewood Cliffs, NJ, 1964.

TSUCHIYA T., **Affine Scaling Algorithm**, in T. Terlaky, (Ed.) Interior Point Methods of Mathematical Programming, KLUWER ACADEMIC PUBLISHERS, Dordrecht, 1996, pp.35-82.

TURNER, K., **Computing Projections for Karmarkar Algorithm**, LINEAR ALGEBRA AND ITS APPLICATIONS, Vol.152, 1991, pp.141-154.

VAIDYA, P.M., **An Algorithm for Linear Programming Which Requires $O((m+n)n^2+(m+n)^{1.5}nL)$**, MATHEMATICAL PROGRAMMING, Vol.47, 1990, pp.175-201.

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

VANDERBEI, R.J., **A Brief Description of ALPO**, Technical Report, AT&T Bell Laboratories, 1990.

VANDERBEI, R.J., **Splitting Dense Columns in Sparse Linear Systems**. LINEAR ALGEBRA AND ITS APPLICATIONS, Vol. 152, 1991, pp.107-117.

VANDERBEI, R.J. and CARPENTER, T.J., **Symmetric Indefinite Systems for Interior Point Methods**, MATHEMATICAL PROGRAMMING, Vol.58, 1993, pp.1-32.

VANDERBEI, R.J., MEKETON, M.S. and FREEDMAN, B.F., **A Modification of Karmarkar's Linear Programming Algorithm**, ALGORITHMICA, Vol.1, 1986, pp.395-407.

WRIGHT, S., **An Interior Point Algorithm for Linearly Constrained Optimization**, SIAM JOURNAL ON OPTIMIZATION, Vol.1, 1992.

WRIGHT, S., **Modified Cholesky Factorizations in Interior Point Algorithms for Linear Programming**, Preprints ANL/MCS - P600-0596, May 1996.

YE, Y., **An O(n^{3}L) Potential Reduction Algorithm for Linear Programming**, MATHEMATICAL PROGRAMMING, Vol.50, 1991, pp.239-258.

ZHANG, Y. and TAPIA, R.A., **A Superlinearly Convergent Polynomial Primal-dual Interior-point Algorithm for Linear Programming**, SIAM JOURNAL ON OPTIMIZATION, Vol.3, 1993, pp.118-133.

ZHANG, Y. , TAPIA, R.A. and DENNIS, J.E., **On the Superlinear and Quadratic Convergence of Primal-dual Interior Point Linear Programming Algorithms**, SIAM JOURNAL ON OPTIMIZATION, Vol.2, no.2, 1992, pp.304-324.-