Revista Colombiana de Matemáticas
Volumen 41 (2007), páginas 197–220

An inexact Newton hybrid path-following algorithm for nonlinear programming

Miguel Argáez
Jaime Hernández Jr.
Leticia Velázquez
The University of Texas, El Paso, Texas, USA

Abstract. In this paper we present a hybrid path-following algorithm that generates inexact Newton steps suited for solving large scale and/or degenerate nonlinear programs. The algorithm uses as a central region a relaxed notion of the central path, called quasicentral path, a generalized augmented Lagrangian function, weighted proximity measures, and a linesearch within a trust region strategy. We apply a semi-iterative method for obtaining inexact Newton steps by using the conjugate gradient algorithm as an iterative procedure. We present a numerical comparison, and some promising results are reported.

Keywords and phrases. Interior-point methods, trust region methods, linesearch technique, nonlinear programming, and conjugate gradient.

2000 Mathematics Subject Classification. Primary: 90C30. Secondary: 90C51, 90C06.

Resumen. En este articulo nosotros presentamos un algoritmo híbrido de seguimiento de camino que genera pasos inexactos de Newton para resolver problemas de gran escala o degenerados para programación no lineal. El algoritmo usa como una region de centralidad una noción mas débil que el bien conocido camino central, llamada camino quasi-central, una generalización de la función aumentada de Lagrange, medidas de aproximación pesadas, y una dirección de búsqueda dentro de una region de verdad. Nosotros aplicamos un método semi-iterativo para obtener direcciones inexactas del método de Newton usando el algoritmo del gradiente conjugado y presentamos una comparación numérica con resultados prometedores.

1. Introduction

The effective development of interior-point methodology for solving linear programs has stirred the study of such methods to the difficult area of non-convex nonlinear programming. In this paper, we implement an inexact Newton hybrid path-following algorithm for solving nonlinear programs. This research has been significantly influenced by the path-following strategy introduced by Argáez [2] and Argáez and Tapia [4]. The algorithm employs the three key elements of this path-following strategy: the quasicentral path, a generalized augmented Lagrangian function, and weighted proximity measures. Moreover, we add a hybrid globalization strategy, and also an inexact Newton step.

The selection of the quasicentral path as a central region, which omits the dual condition from the well known central path, provides some definite advantages. The use of the quasicentral path increases the probability that the central region exists far away from the solution for certain values of the perturbation parameter, when the central path may fail to exist. Furthermore, the central path does not guarantee convergence to an optimal solution of the problem (See Tseng and Ye [19]). It is worth mentioning that the quasicentral path is a variety and not a path. However, we choose to retain the already established terminology originally introduced by Argáez and Tapia.

The use of interior-point methodology for solving large scale problems requires the solution of large, non-symmetric, and highly indefinite linear systems. In this work, the linear systems are decomposed into two smaller subproblems. We apply a trust region strategy for obtaining the solutions of the subproblems. The first subproblem is solved by using a direct dogleg strategy, and the second one by a conjugate gradient algorithm with two extra stopping criteria. One advantage of this procedure is that relaxes the standard Newton assumptions associated with the problem, and increases the likelihood of achieving convergence of degenerate problems. Another advantage is that the two smaller systems can be solved more efficiently in terms of computational time.

After finding an inexact Newton step, and before we test if this is an acceptable direction, we correct the step using a linesearch strategy. This is allowed since we demonstrate that the inexact Newton step is a descent direction for the penalty term of the generalized augmented Lagrangian function. Due to this property, we can select a steplength that generates iterates closer to the quasicentral path, and therefore far away from no-solution boundaries. By incorporating this linesearch strategy, we have increased the likelihood of accepting the corrected inexact Newton step.

Finally, we present a numerical experimentation on a set of problems from the CUTE collection. We perform a numerical comparison of the proposed inexact Newton hybrid path-following algorithm versus the exact Newton hybrid path-following algorithm and the inexact Newton trust region algorithm (i.e. we do not consider the linesearch in the trust region strategy). The numerical results illustrate the viability of the proposed algorithm.

2. Problem formulation

We study the general nonlinear program in the following form:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h(x) = 0 \\ & x \geq 0, \end{array} \tag{2.1}$$

where $h(x) = (h_1(x), \ldots, h_m(x))^T$ and $f, h_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$ $(m \leq n)$ are twice continuously differentiable functions.

The feasible set associated to problem (2.1) is defined as the collection of points that satisfy the equality and inequality constraints; that is, $\Omega = \{x \in \mathbb{R}^n : h(x) = 0, x \geq 0\}$. The corresponding strictly feasible set is denoted by $\Omega^\circ = \{x \in \Omega \text{ and } x > 0\}$.

Definition 2.1. *A solution of problem (2.1) is a vector $x^* \in \Omega$ such that there is a neighborhood U of x^* with $f(x^*) \leq f(x)$ for $x \in U \cap \Omega$.*

The Lagrangian function associated with problem (2.1) is defined as:

$$\ell(x, y, z) = f(x) + h(x)^T y - x^T z,$$

where $y \in \mathbb{R}^m$, and $z \geq 0 \in \mathbb{R}^n$ are the Lagrange multipliers associated with the equality and inequality constraints, respectively.

The first-order necessary conditions of problem (2.1), which are also known as the Karush-Kuhn-Tucker (KKT) conditions, describe a square nonlinear function given by:

$$\begin{gathered} F : \mathbb{R}^{n+m+n} \to \mathbb{R}^{n+m+n} \\ F(x, y, z) \begin{bmatrix} \nabla f(x) + \nabla h(x) y - z \\ h(x) \\ XZe \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ (x, z) \geq 0, \end{gathered} \tag{2.2}$$

where $\nabla h(x) = [\nabla h_1(x), \ldots, \nabla h_m(x)] \in \mathbb{R}^{n \times m}$, $X = \operatorname{diag}(x)$, $Z = \operatorname{diag}(z)$, and $e = (1, \ldots, 1)^T \in \mathbb{R}^n$. The first and second conditions are the partial derivatives of the Lagrangian function with respect to the variables x and y, respectively. The third equation is the complementarity condition that states $x_i z_i = 0$ for $i = 1, \ldots, n$. To find a solution of $F(x, y, z)$, we apply Newton's method due to its fast rate of convergence.

For a feasible point x, we let $\mathcal{B}(x) = \{j : x_j = 0\}$ denote the set of indices of active constraints. The set of active constraint gradients is $\{e_j \in \mathbb{R}^n : j \in \mathcal{B}(x)\}$. Further, the point (x, y, z) is an interior point if $x > 0$ and $z > 0$, which is denoted by $(x, z) > 0$. In particular, interior-point methods keep the iterates strictly positive except possibly at a solution.

2.1. **Standard assumptions.** In the study of Newton's method, the standard assumptions for problem (2.1) are:

A.1: (Existence) There exists a solution x^* to problem (2.1).

A.2: (Smoothness) The Hessian operators $\nabla^2 f,\ \nabla^2 h_i,\ i = 1, \ldots, m$ are Lipschitz continuous in a neighborhood of x^*.

A.3: (Regularity) The set $\{\nabla h_1(x^*), \ldots, \nabla h_m(x^*)\} \cup \{e_j \in \mathbb{R}^n : j \in \mathcal{B}(x^*)\}$ is linearly independent.

A.4: (Second-order sufficiency) For all $\eta \neq 0$ satisfying $\nabla h_i(x^*)^T \eta = 0$, $i = 1, \ldots, m;\ \ e_j^T \eta = 0,\ \ j \in \mathcal{B}(x^*)$, we have:

$$\eta^T \nabla_x^2 \ell(x^*, y^*, z^*)\eta > 0.$$

A.5: (Strict complementarity) For all j, $x_j^* + z_j^* > 0$.

The following relationship between Assumptions A.3-A.5 and the invertibility of the Jacobian matrix of (2.2) can be found in Section 4 of El-Bakry et. al [11].

Proposition 2.2. *Let conditions A.1-A.2 hold. Then the following statements are equivalent:*

(1) *Conditions A.3-A.5 hold.*

(2) *The Jacobian matrix $F'(x^*, y^*, z^*)$ is nonsingular.*

Proof. See El-Bakry et. al [11]. ☑

We say that problem (2.1) is degenerate if Assumptions A.3-A.5 do not hold. This means that at a solution, the Jacobian of F is a singular matrix.

2.2. **Perturbed KKT conditions.** In this section we motivate the use of the perturbed KKT conditions instead of the KKT conditions (2.2) . The complementarity conditions for problem (2.1) are:

$$XZe = 0.$$

By using Newton methodology, we deal with the linearization of the complementarity conditions, given by:

$$Z\Delta x + X\Delta z = -XZe. \tag{2.3}$$

The linearized form of the complementarity condition has a serious flaw. If there is a j such that $x_j = 0$ and $z_j \neq 0$, then from (2.3):

$$\Delta x_j = 0.$$

After the update, the subsequent iterate is:

$$(x_j)_+ = x_j + \alpha \Delta x_j = 0, \quad \text{for any } \ \alpha \in \mathbb{R}.$$

This implies that if a component x_j becomes zero, with its corresponding Lagrange multiplier $z_j \neq 0$, then this component will remain zero for all the subsequent iterations. The analogous situation is also true for the z variable. Such a flaw clearly precludes the global convergence of the algorithm.

We can overcome this difficulty by perturbing the complementarity condition with μe to obtain:

$$XZe = \mu e,\ \mu > 0.$$

In the same situation, it can be shown that $\Delta x_j \neq 0$. Hence, this modification tends to keep the iterates away from the no-solution boundaries.

Then for $\mu \geq 0$, the perturbed KKT conditions of problem (2.1) are given by:

$$F_\mu(x,y,z) = \begin{bmatrix} \nabla f(x) + \nabla h(x)y - z \\ h(x) \\ XZe - \mu e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.4}$$

$$(x,z) \geq 0,$$

where the first, second, and third blocks of equations are called the dual, primal, and perturbed complementarity conditions, respectively.

Now the Newton step $(\Delta x, \Delta y, \Delta z)$ associated to the perturbed KKT conditions is calculated as the exact solution of the following linear system:

$$\begin{bmatrix} \nabla^2_x \ell(x,y,z) & \nabla h(x) & -I \\ \nabla h(x)^T & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = - \begin{bmatrix} e_d \\ e_p \\ e_c \end{bmatrix}, \tag{2.5}$$

where

$$\begin{aligned} \nabla^2_x \ell(x,y,z) &= \nabla^2 f(x) + \textstyle\sum_{j=1}^m \nabla^2 h_j(x) y_j, \\ e_d &\equiv \nabla f(x) + \nabla h(x) y - z, \\ e_p &\equiv h(x), \text{ and} \\ e_c &\equiv XZe - \mu e. \end{aligned}$$

The coefficient matrix of linear system (2.5) is non-symmetric and usually highly indefinite.

Now for a fixed $\mu > 0$, we apply an interior-point Newton method for finding a solution of (2.4). Consequently, we apply a homotopy method for obtaining an optimal solution of (2.1) that must satisfy the KKT conditions (2.2) when $\mu = 0$. In the next section, we present the relevant factors involved in the proposed homotopy method called path-following strategy.

3. Path-following strategy

An important research activity is the development of path-following strategies and its efficient implementations, especially for solving large scale and/or degenerate problems. We follow a path-following strategy introduced by Argáez and Tapia [2, 4] due the promising numerical results obtained in Argáez, Tapia, and Velázquez [5]. In this work, we adapt the strategy using a hybrid linesearch and trust region techniques.

Now we introduce a definition of a path-following strategy presented by Argáez, Tapia, and Velázquez [5].

Definition 3.1. (Path-Following strategy) *For $\mu > 0$, and working from the interior $(x, z) > 0$, apply a globalized Newton's method to the perturbed KKT conditions until the iterate satisfies a specified proximity to the central region. Then decrease μ, specify a new proximity, and repeat the process. Under appropriate conditions, an optimal solution will be obtained as μ approaches zero.*

We now address the three relevant factors involved with an effective path-following strategy: the choices of a central region to follow, a merit function used to make progress towards an optimal solution, and a proximity measure to determine nearness to the central region.

3.1. **Central path and quasicentral path.** The central path associated to problem (2.1) is defined as the set of interior points (x, y, z) that satisfy the perturbed KKT conditions parameterized by $\mu > 0$.

Some researchers have used this notion to promote the global convergence of the interior-point Newton method with partial success [11, 8, 18]. This is because the central path may not exist for some values of the perturbation parameter μ (See [18]). Moreover, even if the central path exists, it is not guaranteed to promote convergence to an optimal solution [19]. To overcome these disadvantages, we use a relaxed notion of the central path known as the quasicentral path introduced by Argáez and Tapia [2, 4]. The quasicentral path excludes the dual condition from the well known central path.

Definition 3.2. *The quasicentral path is defined as the set of points $(x, z) \in \mathbb{R}^{n+n}$ satisfying:*

$$\begin{bmatrix} h(x) \\ XZe - \mu e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad (3.1)$$
$$(x, z) > 0,$$

parameterized by $\mu > 0$.

It is important to mention that the quasicentral path, which describes conditions for $2n$ variables in $2n + m$ dimensional space, is really a surface.

Far from a solution, it may be the case that the point on the central path corresponding to a parameter μ does not exist. Since the quasicentral path is a relaxation of the central path, the likelihood that there is a point for this μ is dramatically improved. It is important to realize that the quasicentral path is equivalent to the strictly feasible set of problem (2.1). Indeed for $\mu > 0$, a point (x, z) is on the quasicentral path, i.e.

$$(x, z) \in S = \{(x, z) \in \mathbb{R}^{n+n} : h(x) = 0,\ XZe - \mu e = 0,\ x, z > 0\},$$

if and only if x is strictly feasible for problem (2.1); that is,

$$x \in \Omega^\circ = \{x \in \mathbb{R}^n :\ h(x) = 0,\ x > 0\}.$$

Therefore, if $\Omega^\circ \neq \emptyset$, then the quasicentral path will always exist for any $\mu > 0$.

3.2. **Merit function.** The merit function plays a fundamental role in any globalization strategy. To ensure that the Newton method converges from far away initial points, a merit function is used to control the size of the steps using a linesearch strategy or to decide whether a step is acceptable in a trust region strategy.

In this paper, we use the generalized augmented Lagrangian function introduced by Argáez and Tapia [2, 4] as a merit function, that is a generalization of the augmented Lagragian function for equality constrained problems presented by Hestenes [13].

This merit function is denoted by M_μ and is defined for any $\mu > 0$ as follows:

$$M_\mu : \mathbb{R}^{n+m+n} \to \mathbb{R}$$

$$M_\mu(x, z; y, \rho) = \ell(x, y, z) + \rho\Phi_\mu(x, z),$$

where $\ell(x, y, z)$ is the Lagrangian function associated with problem (2.1), i.e.

$$\ell(x, y, z) = f(x) + h(x)^T y - x^T z,$$

ρ is a nonnegative penalty parameter, and the penalty term $\Phi_\mu(x, z)$ is given by:

$$\Phi_\mu(x, z) = \frac{1}{2}\, h(x)^T h(x) + x^T z - \mu \sum_{i=1}^{n} \ln (x_i z_i).$$

As shown by Argáez and Tapia [2, 4], the generalized augmented Lagrangian function has the following properties.

Property 3.1. *For $\mu > 0$, if $(x^*_\mu, y^*_\mu, z^*_\mu) > 0$ satisfies the perturbed KKT conditions (2.4), then there is a $\tilde{\rho} > 0$ such that for $\rho > \tilde{\rho}$:*

$$x^*_\mu = \arg\min M_\mu(x, z^*_\mu; y^*_\mu, \rho).$$

Therefore, the merit function retains the stationarity of x^*_μ and adds positive curvature to a solution of (2.4), for appropriately values of the penalty parameter ρ.

It is important to observe that the variable y does not enter into any of the constraints of problem (2.1), and also in the notion of the quasicentral path. Therefore in the merit function the variable y is considered as a parameter, and it is excluded from any descent considerations. However, the dual variable y will still be computed and updated.

The primary role of the penalty term $\Phi_\mu(x, z)$ is to guide the iterates towards the quasicentral path for a given $\mu > 0$. As shown by Argáez and Tapia [2, 4], this progress is facilitated by the following properties:

Property 3.2. *For $\mu > 0$, $\Phi_\mu(x, z)$ is bounded below by its global minimum $n\mu(1 - \log(\mu))$, in the class of all interior points. Furthermore, $\Phi_\mu(x, z)$ assumes this value when (x, z) is on the quasicentral path for a given μ.*

Indeed for $\mu > 0$, the iterates can move towards the quasicentral path, i.e. towards feasibility for problem (2.1), by reducing $\Phi_\mu(x, z)$. It follows that the remaining issue is to determine when the penalty term can be decreased. From a theoretical standpoint, it would be easier to obtain this decrease if the Newton step $(\Delta x, \Delta z)$ was a descent direction for $\Phi_\mu(x, z)$.

Property 3.3. *For $\mu > 0$ and if $(x, z) > 0$ is not on the quasicentral path, then the Newton step $(\Delta x, \Delta z)$ is a descent direction for Φ_μ, i.e.,*

$$\nabla_x \Phi_\mu(x, z)^T \Delta x + \nabla_z \Phi_\mu(x, z)^T \Delta z = -\|h(x)\|^2 + \|W(XZe - \mu e)\|^2 < 0,$$

where $W = (XZ)^{-1/2}$.

Proof. See Argáez and Tapia [2, 4]. ☑

From the two previous properties, when (x, z) is not on the quasicentral path, it is always possible to make progress towards the quasicentral path.

Now the merit function has the following property.

Property 3.4. *For $\mu > 0$ and if (x, z) is not on the quasicentral, then there is a $\tilde{\rho} > 0$ such that for any $\rho > \tilde{\rho}$ any Newton step $(\Delta x, \Delta z)$ is a descent direction for the function M_μ, i.e.*

$$\nabla_x M_\mu(x, z; y, \rho)^T \Delta x + \nabla_z M_\mu(x, z; y, \rho)^T \Delta z < 0.$$

Proof. See Argáez and Tapia [2, 4]. ☑

Now, it is important to observe that Properties 1 and 4 of the merit function M_μ are dependent upon the selection of the penalty parameter ρ. In section 5, we present our procedure for determining ρ.

3.3. **Weighted proximity measure.** In the implementation of a path-following strategy, the choice of a proximity measure to the central region can affect the performance of any interior-point algorithm. This is an important issue since working to satisfy strict proximity measures to the quasicentral path can make the procedure computationally expensive. On the contrary, if we work towards more lenient proximity measures, then the global convergence can be threatened by the no-solution boundaries. To measure nearness to the quasicentral path for a fixed μ, we use the following neighborhood denoted by N_w, presented by Argáez and Tapia [2, 4]:

$$N_W(\gamma\mu) = \{(x, y, z) \in \mathbb{R}^{n+m+n} : \|h(x)\|^2 + \|W(XZe - \mu e)\|^2 \leq \gamma\mu\},$$

where $W = (XZ)^{-\frac{1}{2}}$ and $\gamma \in (0, 1]$.

Therefore for a $\mu > 0$, we consider an interior point $(x, y, z) \in N_W(\gamma\mu)$ to be close to the quasicentral path. Notice that the first and second terms of the inequality measure infeasibility of the equality constraints, and the perturbed complementarity condition weighted with the positive definite matrix W, respectively. A brief theoretical comparison between weighted and nonweighted proximity measures is given in Argáez, Mendez, and Velázquez [7].

4. Inexact Newton steps

In the path-following strategy described before, standard Newton assumptions are considered. However, for some real scientific applications these assumptions are not met and/or the size of the problem makes it difficult for computing Newton steps. To overcome these disadvantages, one of the goals of this paper is to present a new strategy for obtaining inexact Newton steps that fits with the path-following strategy and allows to obtain an optimal solution by relaxing the standard conditions and efficiently from computational point of view.

4.1. **Reduced system.** To take advantage of the structure of the problem, we decouple the perturbed KKT system. From the third block of equations of (2.5) we have:

$$\Delta z = -X^{-1}(e_c + Z\Delta x). \tag{4.1}$$

By substituting (4.1) into system (2.5), we obtain a reduced system of equations, which is known as the augmented system:

$$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = - \begin{bmatrix} c \\ e_p \end{bmatrix} \tag{4.2}$$

where

$$\begin{aligned} Q &\equiv \nabla_x^2 \ell(x, y, z) + X^{-1}Z, \\ A &\equiv \nabla h(x)^T, \text{ and} \\ c &\equiv e_d + X^{-1}e_c. \end{aligned}$$

The augmented system (4.2) can be considered as the necessary condition for Δx to be a solution of the following quadratic problem:

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\Delta x^T Q \Delta x + c^T \Delta x \\ \text{subject to} \quad & A\Delta x + e_p = 0, \end{aligned} \tag{4.3}$$

with Q, A, and c defined in (4.2). It is well known that if Q is positive definite on the null space of A, denoted as $Q > 0$ on $\mathcal{N}(A)$, and A is a full-rank matrix, then the quadratic problem (4.3) has a unique global solution Δx. This solution can be obtained by solving the augmented system (4.2) where Δx is the solution of the problem and Δy is the Lagrange multiplier associated with the equality constraint.

For our purpose, this reduced system improves the likelihood that Q is positive definite on $\mathcal{N}(A)$ because of the addition of the positive definite diagonal matrix $X^{-1}Z$ to the second order information associated to the problem. This strategy allows us to relax the standard Assumption A.4.

Next, we describe a general formulation for obtaining a (approximate) solution of problem (4.3) with a low computational cost.

Since A is a linear operator from $\mathbb{R}^n$ to $\mathbb{R}^m$, then Δx can be expressed as a direct sum of one element in the row space of A, $\Delta x_p \in \mathcal{R}(A^T)$, and another in the null space of A, $\Delta x_h \in \mathcal{N}(A)$; that is,

$$\Delta x = \Delta x_p + \Delta x_h,$$

where Δx_p and Δx_h are orthogonal. We call Δx_p and Δx_h the particular and homogeneous solutions, respectively.

If $e_p \in \mathcal{R}(A)$, as a consequence of the fundamental theorem of linear algebra, there exists a unique particular solution $\Delta x_p \in \mathcal{R}(A^T)$ that is the minimum norm solution of the under-determined linear constraint $A\Delta x + e_p = 0$. This particular solution is given by

$$\Delta x_p = -A^\dagger e_p,$$

where $A^\dagger$ is the Moore-Penrose pseudoinverse of A. By the standard Newton Assumption A.3, A is a full-rank matrix, and therefore the pseudoinverse is given by

$$A^\dagger = A^T(AA^T)^{-1}.$$

In the case the Assumption A.3 is not met, then the particular solution is also characterized in terms of the Moore-Penrose pseudoinverse. This particular solution is known as the solution of the following least squares minimum norm problem:

$$\begin{array}{ll} \text{minimize} & \|\Delta x\| \\ \text{subject to} & A\Delta x + e_p = 0. \end{array} \tag{4.4}$$

Once $\Delta x_p \in \mathcal{R}(A^T)$ is found, and since no further progress can be made towards satisfying the equality constraint of (4.3), we concentrate on minimizing the objective function of problem (4.3) with respect to the homogeneous solution Δx_h. By doing so, we obtain Δx_h as a solution of the following constrained quadratic minimization problem:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\Delta x^T Q \Delta x + (Q\Delta x_p + c)^T \Delta x \\ \text{subject to} & \Delta x \in \mathcal{N}(A). \end{array} \tag{4.5}$$

This problem can be modeled as an unconstrained minimization quadratic problem by substituting Δx_h for Pw, where P is an orthogonal projector onto $\mathcal{N}(A)$, and $w \in \mathbb{R}^n$. That is:

$$\text{minimize} \quad \tfrac{1}{2} w^T (PQP) w + (P(Q\Delta x_p + c))^T w. \tag{4.6}$$

In the case A is full rank, the projection is given by $P = I - A^T(AA^T)^{-1}A$.

Now we obtain Δx_p and Δx_h by adding a trust region constraint to their associated subproblems (4.4) and (4.6). The addition of this constraint improves the likelihood of finding good approximate solutions to the two subproblems, and increases the numerical stability and robustness of the linear algebra routines involved. Most importantly, this strategy calculates an approximate solution of problem (4.3), and equivalently system (4.2), even when the standard assumptions associated to (2.1) do not hold. Now, we present the techniques for obtaining Δx_p and Δx_h that fit with the path-following strategy.

4.2. **Particular solution.** The particular solution Δx_p is obtained as the solution of the following constrained least squares problem:

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\|A\Delta x_p + e_p\|^2 \\ \text{subject to} \quad & \Delta x_p \in \mathcal{R}(A^T) \\ & \|\Delta x_p\| \leq \delta\Delta, \end{aligned} \tag{4.7}$$

where $\delta \in (0,1)$ and $\Delta > 0$ is known as the trust region radius. Upon the expansion of the objective function, we obtain the following quadratic problem:

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\Delta x^T A^T A\Delta x + (A^T e_p)^T \Delta x \\ \text{subject to} \quad & \Delta x \in \mathcal{R}(A^T) \\ & \|\Delta x\| \leq \delta\Delta. \end{aligned} \tag{4.8}$$

We find Δx_p of (4.8) by modifying the single dogleg algorithm [9, 16]. The two principal components of dogleg algorithm are the Cauchy point and Newton step given by the following definitions.

Definition 4.1. *The Cauchy point is defined as:*

$$\Delta x_p^{\mathrm{CP}} = -\frac{\|A^T e_p\|^2}{\|AA^T e_p\|^2} A^T e_p.$$

Definition 4.2. *The Newton step is defined as:*

$$\Delta x_p^{\mathrm{N}} = -A^\dagger e_p = -A^T (AA^T)^{-1} e_p,$$

where $A^\dagger$ is Moore-Penrose pseudoinverse of A.

By using these definitions, we present an algorithm for obtaining the particular solution Δx_p:

Algorithm 4.1. (Modified dogleg algorithm)
Given $\Delta > 0$ and $\delta \in (0,1)$, do the following:

Step 1: *Compute Δx_p^{CP} and Δx_p^{N} using Definitions 4.1 and 4.2.*

Step 2: *If $(\|\Delta x_p^{\mathrm{CP}}\| > \delta\Delta$ and $\|\Delta x_p^{\mathrm{N}}\| > \delta\Delta)$, then:*

2a: *Compute the scaled versions of Δx_p^{CP} and Δx_p^{N}:*

$$\Delta x_p^{\mathrm{CP}} = -\frac{\delta\Delta}{\|A^T e_p\|} A^T e_p \;\text{ and }\; \Delta x_p^{\mathrm{N}} = -\frac{\delta\Delta}{\|A^\dagger e_p\|} A^\dagger e_p.$$

2b: *Choose Δx_p to minimize the residual: $\|A\Delta x_p + e_p\|$. That is, choose Δx_p such that:*

$$\|A\Delta x_p + e_p\| = min\{\|A\Delta x_p^{\mathrm{CP}} + e_p\|, \|A\Delta x_p^{\mathrm{N}} + e_p\|\}.$$

Step 3: *If $(\|\Delta x_p^{\mathrm{CP}}\| > \delta\Delta$ and $\|\Delta x_p^{\mathrm{N}}\| \leq \delta\Delta)$, then $\Delta x_p = \Delta x_p^{\mathrm{N}}$.*

Step 4: *If $(\|\Delta x_p^{\mathrm{CP}}\| \leq \delta\Delta$ and $\|\Delta x_p^{\mathrm{N}}\| > \delta\Delta)$, then:*

4a: *Find $\alpha \geq 0$ such that $\Delta x_p^{\mathrm{DOG}} = \Delta x_p^{\mathrm{CP}} + \alpha(\Delta x_p^{\mathrm{N}} - \Delta x_p^{\mathrm{CP}})$ satisfies $\|\Delta x_p^{\mathrm{DOG}}\| = \delta\Delta$.*

4b: *Compute the scaled version of Δx_p^{N}, as in 2a.*

4c: *Choose Δx_p to minimize the residual between $\Delta x_p^{\mathrm{DOG}}$ and Δx_p^{N}, as in 2b.*

Step 5: *If ($\|\Delta x_p^{\mathrm{CP}}\| \leq \delta\Delta$ and $\|\Delta x_p^{\mathrm{N}}\| \leq \delta\Delta$), then choose Δx_p to be the largest step in norm, that is, choose $\|\Delta x_p\| = max\{\|\Delta x_p^{\mathrm{CP}}\|, \|\Delta x_p^{\mathrm{N}}\|\}$.*

Upon the completion of Algorithm 4.1, we compute the error of the particular solution r_{ps} as follows: $r_{\mathrm{ps}} \equiv A\Delta x_p + e_p$.

It is well known that the primary advantage of the traditional single dogleg algorithm is its ability to produce fairly accurate solutions of problem (4.8), while incurring relatively low computational costs. Most importantly, there are cases when the single dogleg algorithm avoids the computationally expensive formation of $A^\dagger = A^T(AA^T)^{-1}$.

Regardless of whether or not $A^\dagger$ was used in the computation of Δx_p, it will be needed for obtaining the homogeneous solution Δx_h by using the projection operator $P = I - A^\dagger A$. As a result, any savings in computational costs from using the single dogleg algorithm will be squandered by the computation of Δx_h. Then, we incorporate the computation of $A^\dagger$ into the traditional single dogleg algorithm to form Algorithm 4.1.

Within Algorithm 4.1, due to the constraint, it was often necessary to scale Δx_p^{CP} and Δx_p^{N} to satisfy the trust region bound. In this direction, we define the Cauchy and Newton steps associated to (4.8): the two vital components of the modified dogleg algorithm.

Definition 4.3. *The constrained Cauchy step is defined as:*

$$\Delta x_p^{\mathrm{CP}} = \begin{cases} -\frac{\delta\Delta}{\|A^T e_p\|} A^T e_p, & if \;\; (A^T e_p)^T A^T A (A^T e_p) \leq 0 \;\; or \;\; \frac{\|A^T e_p\|^3}{\|AA^T e_p\|^2} > \delta\Delta, \\ -\frac{\|A^T e_p\|^2}{\|AA^T e_p\|^2} A^T e_p, & otherwise. \end{cases}$$

Definition 4.4. *The constrained Newton step is defined as:*

$$\Delta x_p^{\mathrm{N}} = \begin{cases} -\frac{\delta\Delta}{\|A^\dagger e_p\|} A^\dagger e_p, & if \;\; \|A^\dagger e_p\| > \delta\Delta, \\ -A^\dagger e_p, & otherwise. \end{cases}$$

4.3. **Homogeneous solution.** The homogeneous solution $\Delta x_h = Pw$ is obtained from the solution of the following constrained minimization problem:

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2} w^T PQPw + (P(Q\Delta x_p + c))^T w \\ \text{subject to} \quad & \|Pw\| \leq \overline{\Delta} \end{aligned} \tag{4.9}$$

where $\overline{\Delta} = \sqrt{\Delta^2 - \|\Delta x_p\|^2}$.
Finding a solution of the objective function of problem (4.9) is equivalent to solving the following constrained linear system:

$$PQPw = -P(Q\Delta x_p + c). \tag{4.10}$$

Equation (4.10) is a consistent equation since the operator P appears in both left sides. Since P is a singular matrix, therefore this system of equations has infinite solutions. Now a solution of (4.10) such that PQP is at least positive

semidefinite in the $\mathcal{N}(A)$ is a solution for the objective function of problem (4.9).

Equation (4.10) is well suited to be solved by applying the conjugate gradient algorithm, and using as an initial approximation any value in the $\mathcal{N}(A)$. In particular, we use the initial approximation vector $w = 0$. Under these conditions, the algorithm generates sequences of approximate solutions, residuals and directions in the $\mathcal{N}(A)$. Moreover, by choosing the initial value of $w = 0$, then the norm of the sequences of approximate solutions are increasing. Therefore, this procedure allows to find a solution of problem (4.9) by applying the conjugate gradient algorithm to problem (4.10). In this way, the ellipsoidal constraint ($\|Pw\| \leq \overline{\Delta}$) of problem (4.9) is deactivated and converted into a spherical constraint ($\|w\| \leq \overline{\Delta}$).

For solving problem (4.10), we need to consider two extra stopping termination tests. The first test consists when the norm of the approximate solution violates the trust region radius for problem (4.9), and the second one is when Q is not positive definite on $\mathcal{N}(A)$ (See [2]). Furthermore, since the number of eigenvalues are at most $n - m$, then the conjugate gradient converges in no more than $n - m$ iterations in infinite arithmetic.

Algorithm 4.2. (Conjugate Gradient Algorithm)
Given $\epsilon_1 > 0$, $I_{\text{cg}} > 0$, $w_0 = 0$, $r_0 = P(Q\Delta x_p + c)$, *and* $d_0 = -r_0$.
For $k = 0, 1, 2, \ldots, I_{\text{cg}} > 0$, *do the following:*

Step 1: *Compute* $q_k = PQd_k$.

Step 2: *If* $d_k^T q_k \leq 0$, *find an* $\alpha \geq 0$ *such that* $w = w_k + \alpha d_k$ *minimizes (4.9) and satisfies* $\|w\| = \overline{\Delta}$. *STOP.*

Step 3: *Compute* $\alpha_k = \frac{r_k^T r_k}{d_k^T Q d_k}$ *and* $w_{k+1} = w_k + \alpha_k d_k$.

Step 4: *If* $\|w_{k+1}\| > \overline{\Delta}$, *find an* $\alpha \geq 0$ *such that* $w = w_k + \alpha d_k$ *satisfies* $\|w\| = \overline{\Delta}$. *STOP.*

Step 5: *Compute* $r_{k+1} = r_k + \alpha_k q_k$.

Step 6: *If* $\|r_{k+1}\| \leq \epsilon$, $w = w_{k+1}$. *STOP.*

Step 7: *Compute* $\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$, $d_{k+1} = r_{k+1} + \beta_{k+1} d_k$, *update* $k = k + 1$, *and go to Step 1.*

Upon the completion of Algorithm 4.2, we compute the error of the homogeneous solution r_{cg} as follows $r_{\text{cg}} \equiv P(Q(w + \Delta x_p) + c)$.

4.4. **Inexact Newton step.** We form Δx after computing Δx_p and Δx_h from Algorithms 4.1 and 4.2, respectively. Then, we obtain Δy and Δz.

Definition 4.5. *An inexact Newton step* $(\Delta x, \Delta y, \Delta z)$ *for system (2.5) is given by*

$$\begin{aligned} \Delta x &= \Delta x_p + \Delta x_h \\ \Delta y &= (AA^T)^{-1} A(Q\Delta x + c) \\ \Delta z &= -X^{-1}(e_c + Z\Delta x), \end{aligned}$$

where Δx_p *and* Δx_h *are obtained using Algorithms 4.1 and 4.2, respectively.*

Notice that we are relaxing the standard Assumptions A.3-A.5., i.e., the inexact Newton step is calculated independent of the invertibility of the coefficient matrix in (2.5). In addition, since we are solving two smaller subproblems, then the computation of the inexact step is less costly when compared to the exact step.

Since the inexact Newton step is an approximate solution of (2.5), we can determine the resulting deviation from its exact counterpart. In the following proposition, we show that the error between the inexact and exact Newton steps depends only on the errors of the particular and homogeneous solutions, r_{ps} and r_{cg}, respectively.

Proposition 4.6. *The inexact Newton step* $(\Delta x, \Delta y, \Delta z)$ *given by Definition 4.5 satisfies*

$$\begin{bmatrix} \nabla_x^2 \ell(x,y,z) & \nabla h(x) & -I \\ \nabla h(x)^T & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = - \begin{bmatrix} e_d \\ e_p \\ e_c \end{bmatrix} + \begin{bmatrix} r_{\text{cg}} \\ r_{\text{ps}} \\ 0 \end{bmatrix}. \tag{4.11}$$

Proof. Using the projector $P = I - A^\dagger A$, we obtain

$$r_{\text{cg}} = P(Q\Delta x + c) = (Q\Delta x + c) - A^\dagger A(Q\Delta x + c) = (Q\Delta x + c) + A^T \Delta y.$$

From (4.2) and the third block of equations of linear system (2.5), then

$$\nabla_x^2 \ell(x,y,z)\Delta x + \nabla h(x)\Delta y - \Delta z = -e_d + r_{\text{cg}}.$$

Now, since $\Delta x = \Delta x_p + w$, we have that

$$\nabla h(x)^T \Delta x = \nabla h(x)^T (\Delta x_p + w) = \nabla h(x)^T \Delta x_p = -e_p + r_{\text{ps}}.$$

☑

Therefore, by controlling the errors associated with the particular and homogeneous solutions, we control the deviation between the inexact and exact Newton steps. When the iterates of the procedure move closer to a solution, the error due to the particular solution usually vanishes and the total error of the inexact Newton step depends upon the error of the conjugate gradient algorithm. In particular, if $r_{\text{ps}} = 0$ and $r_{\text{cg}} = 0$, then the inexact and exact Newton steps will be equivalent.

In the last section, we discussed the descent properties of $\Phi_\mu(x,z)$ when using the exact Newton step. Now we show that the descent properties of $\Phi_\mu(x,z)$ are maintained when using the inexact Newton step proposed by Definition 4.5.

When computing the directional derivative of $\Phi_\mu(x,z)$, with respect to the inexact Newton step, we can no longer make the assumption that $A\Delta x_p = -e_p$, as was the case in Property 3.3. We will have to determine the effect that r_{ps} has on the computation of the derivative. Since r_{ps} is dependent of Δx_p, we need to consider the five possible outcomes for Δx_p when using Algorithm 4.1 listed in Table 1. In the following proposition, as the other cases generalize from Δx_p^{DOG}, we explicitly compute the directional derivative of $\Phi_\mu(x,z)$ when

TABLE 1. Possible Options for Δx_p.

1	$\Delta x_p = -\frac{\delta\Delta}{\|A^T e_p\|} A^T e_p$
2	$\Delta x_p = -\frac{\delta\Delta}{\|A^\dagger e_p\|} A^\dagger e_p$
3	$\Delta x_p = -A^\dagger e_p$
4	$\Delta x_p = -\frac{\|A^T e_p\|^2}{\|AA^T e_p\|^2} A^T e_p$
5	$\Delta x_p = \Delta x_p^{\mathrm{DOG}}$

$\Delta x_p = \Delta x_p^{\mathrm{DOG}}$, in the direction of the inexact step $(\Delta x, \Delta z)$. The other cases will follow similarly.

Proposition 4.7. *For $\mu > 0$ and if $(x, z) > 0$ is not quasicentral, then the inexact Newton step $(\Delta x, \Delta z)$ is a descent direction for Φ_μ, i.e.*

$$\nabla_x \Phi_\mu(x,z)^T \Delta x + \nabla_z \Phi_\mu(x,z)^T \Delta z < 0.$$

Proof. The gradients of $\Phi_\mu(x, z)$ with respect to x and z are:

$$\nabla_x \Phi_\mu(x,z) = \nabla h(x) h(x) + z - \mu X^{-1} e$$
$$\nabla_z \Phi_\mu(x,z) = x - \mu Z^{-1} e.$$

By using the inexact step, the directional derivative of $\Phi_\mu(x, z)$ is computed as:

$$\nabla_x \Phi_\mu(x,z)^T \Delta x + \nabla_z \Phi_\mu(x,z)^T \Delta z = (e_p^T A + z^T - \mu e^T X^{-1})\Delta x + (x^T - \mu e^T Z^{-1})\Delta z.$$

If we set $W = (XZ)^{-\frac{1}{2}}$, after some algebraic operations, we obtain the following expression:

$$\nabla_x \Phi_\mu(x,z)^T \Delta x + \nabla_z \Phi_\mu(x,z)^T \Delta z = e_p^T A \Delta x + \|W(XZe - \mu e)\|^2.$$

Since $A\Delta x = A\Delta x_p = r_{\mathrm{ps}} - e_p$, this equation becomes:

$$\nabla_x \Phi_\mu(x,z)^T \Delta x + \nabla_z \Phi_\mu(x,z)^T \Delta z = e_p^T (r_{\mathrm{ps}} - e_p) + \|W(XZe - \mu e)\|^2.$$

By working under the assumption that $\Delta x_p = \Delta x_p^{\mathrm{DOG}}$, computed as in Step 4a of Algorithm 4.1, we compute r_{ps} as follows:

$$r_{\mathrm{ps}} = (1-\alpha)(I_{(m\times m)} - \frac{\|A^T e_p\|^2}{\|AA^T e_p\|^2} AA^T) e_p.$$

By substituting this instance of the error due to the particular solution into the directional derivative, we obtain:

$$-(\alpha\|h(x)\|^2 + (1-\alpha)\frac{\|A^T e_p\|^4}{\|AA^T e_p\|^2} + \|W(XZe - \mu e)\|^2) < 0,$$

because $\alpha \in (0, 1)$. ☑

TABLE 2. Directional Derivatives of $\Phi_\mu(x,z)$ for varying Δx_p

1	$-(\frac{\delta\Delta}{\|A^T h(x)\|}\|A^T h(x)\|^2 + \|W(XZe-\mu e)\|^2) < 0$
2	$-(\frac{\delta\Delta}{\|A^\dagger h(x)\|}\|h(x)\|^2 + \|W(XZe-\mu e)\|^2) < 0$
3	$-(\|h(x)\|^2 + \|W(XZe-\mu e)\|^2) < 0$
4	$-(\frac{\|A^T h(x)\|^4}{\|AA^T h(x)\|^2} + \|W(XZe-\mu e)\|^2) < 0$

Remark 4.1. *To complete the proof of Proposition 4.7, we present the four remaining directional derivatives of Φ_μ in Table 2.*

As a result, we can always decrease Φ_μ which allows to make progress towards the quasicentral path. Due to Proposition 4.7, the inexact Newton step also is a descent direction for the merit function M_μ for sufficiently large ρ.

Proposition 4.8. *For $\mu > 0$ and if $(x,z) > 0$ is not on the quasicentral path, then there is a $\tilde{\rho}$ such that for any $\rho > \tilde{\rho}$, the inexact Newton step $(\Delta x, \Delta z)$ is a descent direction for M_μ, i.e.*

$$\nabla_x M_\mu(x,z;y,\rho)^T \Delta x + \nabla_z M_\mu(x,z;y,\rho)^T \Delta z < 0.$$

Proof. Analogous to Proposition 4.7, we compute the gradients of $M_\mu(x,z;y,\rho)$ with respect to x and z to obtain:

$$\begin{aligned}\nabla_x M_\mu(x,z;y,\rho) &= \nabla_x \ell(x,y,z) + \rho \nabla_x \Phi_\mu(x,z)\\ \nabla_z M_\mu(x,z;y,\rho) &= \nabla_z \ell(x,y,z) + \rho \nabla_z \Phi_\mu(x,z).\end{aligned}$$

The directional derivative of $M_\mu(x,z;y,\rho)$ in the direction of the inexact step $(\Delta x, \Delta z)$ is computed as:

$$\begin{aligned}\nabla_x M_\mu(x,z;y,\rho)^T \Delta x + \nabla_z M_\mu(x,z;y,\rho)^T \Delta z &= \nabla_x \ell(x,y,z)^T \Delta x\\ +\nabla_z \ell(x,y,z)^T \Delta z + \rho(\nabla_x \Phi_\mu(x,z)^T \Delta x &+ \nabla_z \Phi_\mu(x,z)^T \Delta z).\end{aligned}$$

Then

$$\nabla_x M_\mu(x,z;y,\rho)^T \Delta x + \nabla_z M_\mu(x,z;y,\rho)^T \Delta z < 0,$$

for any

$$\rho > \frac{\nabla_x \ell(x,y,z)^T \Delta x + \nabla_z \ell(x,y,z)^T \Delta z}{|\nabla_x \Phi_\mu(x,z)^T \Delta x + \nabla_z \Phi_\mu(x,z)^T \Delta z|}$$

because

$$\nabla_x \Phi_\mu(x,z)^T \Delta x + \nabla_z \Phi_\mu(x,z)^T \Delta z < 0.$$

☑

4.5. **Trust region model.** In a trust region globalization strategy, we accept the inexact Newton step $(\Delta x_k, \Delta z_k)$ if the ratio between the merit function and an approximation model, usually denoted by σ_k, is greater than or equal to some η where $\eta \in [\frac{1}{4}, 1]$. As in the thesis of El-Alem [10], our model is comprised of the second and first-order Taylor expansions of $\ell(x, y, z)$ and $\Phi_\mu(x, z)$ with respect to x and z, respectively. Let us denote $x_{k+1} = x_k + \alpha_k \Delta x_k$ and $z_{k+1} = z_k + \alpha_k \Delta z_k$.

Then, we predict the behavior of the merit function at the $k-$th iteration with the following model

$$\hat{M}_\mu(x_{k+1}, z_{k+1}; y_k, \rho) \equiv \hat{\ell}(x_{k+1}, y_k, z_{k+1}) + \rho \Phi_\mu(x_{k+1}, z_{k+1}) \tag{4.12}$$

where

$$\hat{\ell}(x_{k+1}, y_k, z_{k+1}) \equiv \ell(x_k, y_k, z_k) + \alpha_k \nabla f(x_k)^T \Delta x_k + \alpha y_k^T \nabla h(x_k)^T \Delta x_k -$$

$$\alpha_k z_k^T \Delta x_k - \alpha_k x_k^T \Delta z_k - \alpha_k^2 \Delta {x_k}^T \Delta z_k + \frac{\alpha_k^2}{2} \Delta {x_k}^T \nabla_x^2 \ell(x_k, y_k, z_k) \Delta x_k$$

and

$$\hat{\Phi}_\mu(x_{k+1}, z_{k+1}) \equiv \Phi_\mu(x_k, z_k) - \alpha \mid \nabla_x \Phi_\mu(x_k, z_k)^T \Delta x + \nabla_z \Phi_\mu(x_k, z_k)^T \Delta z \mid .$$

From (4.12), we compute σ_k as follows [9, 16]:

$$\sigma_k = \frac{\Delta M_\mu}{\Delta \hat{M}_\mu} = \frac{M_\mu(x_{k+1}, z_{k+1}; y_k, \rho) - M_\mu(x_k, z_k; y_k, \rho)}{\hat{M}_\mu(x_{k+1}, z_{k+1}; y_k, \rho) - \hat{M}_\mu(x_k, z_k; y_k, \rho)}. \tag{4.13}$$

We accept $(\Delta x_k, \Delta z_k)$ as an update if $\sigma_k \geq \eta$; otherwise, we reduce the trust region radius Δ, compute a new inexact Newton step, and repeat the process (See [9, 16]).

5. Algorithm

In this section, we present an inexact Newton path-following interior-point algorithm for nonlinear programming. First, we describe the development of a hybrid component that combines a linesearch within a trust region globalization strategy. Second, we present our selection of the penalty parameter ρ in the algorithm. Finally, we present the algorithm, a numerical experimentation that validates the proposed algorithm.

5.1. **Hybrid globalization strategy.** By using a trust region strategy, we form an inexact Newton step $(\Delta x_k, \Delta y_k, \Delta z_k)$ as an approximate solution of linear system (2.5). As discussed previously, the purposes of the trust region ratio σ_k (4.13) are to determine whether or not to accept the inexact step, modify the trust region radius, and update the current iterate.

Furthermore, we may correct the inexact Newton step to guarantee that the next iterate is also an interior point. Working from an interior point (x_k, y_k, z_k), we perform this correction by choosing a steplength $\alpha_k^+ \in (0, 1]$ such that:

$$(x_{k+1}, z_{k+1}) = (x_k + \alpha_k^+ \Delta x_k, z_k + \alpha_k^+ \Delta z_k) > 0. \tag{5.1}$$

We choose α_k^+ as follows

$$\alpha_k^+ = \min(1, \tau\widehat{\alpha}), \tag{5.2}$$

where

$$\widehat{\alpha} = \min\left(\frac{-1}{\min(X_k^{-1}\Delta x_k, -1)}, \frac{-1}{\min(Z_k^{-1}\Delta z_k, -1)}\right)$$

for some $\tau \in (0,1)$.

In the last section, we showed that the inexact Newton step is a descent direction for Φ_μ when (x,z) is not on the quasicentral path. To take advantage of this property, we implement a linesearch strategy, described in the next algorithm, to compute a steplength α_k that satisfies Armijo's sufficient decrease condition for Φ_μ.

Algorithm 5.1. (Linesearch strategy)
Given α_k^+ in equation (5.2), $s = 1/2$, and $\beta \in (0,1)$, for $j = 1, \ldots,$ do the following:

Step 1: *Calculate $\alpha_k = s^j \alpha_k^+$.*
Step 2: *Compute $(x_{k+1}, z_{k+1}) = (x_k + \alpha_k \Delta x_k, z + \alpha_k \Delta z_k)$.*
Step 3: *If $\Phi_\mu(x_{k+1}, z_{k+1}) \leq \Phi_\mu(x_k, z_k) - 10^{-4}\alpha_k \mid \nabla_x \Phi_\mu(x_k, z_k)^T \Delta x_k + \nabla_z \Phi_\mu(x_k, z_k)^T \Delta z_k \mid$,*
Stop; else $j = j + 1$.

By incorporating Algorithm 5.1 into the trust region framework, we have created a hybrid algorithm. The procedure uses trust region globalization to compute the inexact Newton step and a linesearch backtracking strategy to compute the steplength.
Such a modification assists in obtaining a strictly feasible iterate that is closer to the quasicentral path, and keeps iterates away from no-solution boundaries.

5.2. **Update of the penalty parameter.** Since the acceptance of the inexact Newton step depends on the behavior of the model $\hat{M}_\mu$, then we choose ρ to guarantee a decrease of the model, i.e.,

$$\begin{aligned}\nabla \hat{M}_\mu{}^T v_k =& \nabla \ell(v_k)^T \Delta \tilde{v}_k + \alpha_k \Delta x_k^T \left(\frac{1}{2}\nabla_x^2 \ell(v_k)\Delta x_k - \Delta z_k\right) \\ & - \rho \mid \nabla \Phi_\mu(\tilde{v}_k)^T \Delta \tilde{v}_k \mid < 0\end{aligned}$$

where $v_k = (x_k, y_k, z_k)$, $\tilde{v}_k = (x_k, z_k)$ and $\Delta\tilde{v}_k = (\Delta x_k, \Delta z_k)$.

In the above equation, we treat the variable y essentially as a parameter, i.e., we do not differentiate the model $\hat{M}_\mu$ with respect to y and it is excluded from our descent consideration.

To force a descent direction for the approximated model, i.e., to satisfy the previous inequality, we propose the following algorithm:

Algorithm 5.2. (Selection of penalty parameter)
Given a constant c, do the following:

Step 1: *Compute* $\tilde{\rho} = \rho_1 + \rho_2$
where $\rho_1 = \frac{\nabla \ell(v_k)^T \Delta \tilde{v}_k}{|\nabla \Phi_\mu(\tilde{v}_k)^T \Delta \tilde{v}_k|}$ *and* $\rho_2 = \frac{1/2 \Delta x_k^T (\nabla_x^2 \ell(v_k) \Delta x_k - \Delta z_k)}{|\nabla \Phi_\mu(\tilde{v}_k)^T \Delta \tilde{v}_k|}$.
Step 2: *Compute* $\rho = \max\{0, \tilde{\rho} + c\}$.

While it seems that this selection of ρ only produces a decrease of the model, the use of Algorithm 5.2 also converts the inexact Newton step into a descent direction for M_μ.

As shown by Argáez and Tapia [4], for any $\rho \geq \rho_1$, then the inexact direction (x_k, z_k) is a descent direction for M_μ. If $\rho_2 \geq 0$, it follows that $\rho > \rho_1$ and the inexact Newton direction is also a descent direction for M_μ.

In the case that α_k^+ in equation (5.2) does not allow a reduction of M_μ with the updated iterate and $\rho \geq \rho_1$, then the use of the modified α_k given by Algorithm 5.1 improves the chances of decreasing M_μ and accepting the inexact Newton step.

5.3. **Algorithm.** We now present our inexact Newton hybrid path-following algorithm for nonlinear programming.

Algorithm 5.3. (Inexact Newton hybrid path-following algorithm) *Given* $\epsilon > 0$, $k_{max} > 0$, *an initial interior point* (x_0, y_0, z_0), $\Delta_0 > 0$, $\eta \in [\frac{1}{4}, 1]$, *and* $\tau, \beta, \gamma \in (0, 1]$, *for* $k = 0, 1, 2, \ldots, k_{max}$, *do the following:*

Step 1: *Compute* μ_k.
Step 2: *(Inner Loop) Repeat.*
2a: *Compute the inexact Newton step* $(\Delta x_k, \Delta y_k, \Delta z_k)$ *with Definition 4.5.*
2b: *(Enforce Positivity) Compute* α_k^+ *via (5.2).*
2c: *(Armijo's Sufficient Decrease of* Φ_μ*) Compute* α_k *with Algorithm 5.1.*
2d: *Compute* ρ_k *with Algorithm 5.2.*
2e: *If* $\sigma_k(\alpha_k) \geq \eta$, *update* Δ_k *and accept the inexact Newton step:*

$$(x_{k+1}, y_{k+1}, z_{k+1}) = (x_k + \alpha_k \Delta x_k, y_k + \alpha_k \Delta y_k, z_k + \alpha_k \Delta z_k).$$

Else, update Δ_k, $k = k + 1$, *and go to Step 2.*
Step 3.: *Proximity to the Quasicentral Path.*
If $(x_{k+1}, z_{k+1}) \notin N_W(\gamma \mu_k)$, *update* $k = k + 1$ *and go to Step 2.*
Else if $\|F(x_{k+1}, y_{k+1}, z_{k+1})\| \leq \epsilon$, *STOP.*
Else, update $k = k + 1$ *and go to Step 1.*

5.4. **Numerical experimentation.** Algorithm 5.3 was implemented on a SPARC Station 5 running the SunOS operating system, release 4.1, with 64 megabytes of memory and was written in MATLAB version 6.5. The numerical experimentation was performed on a set of test problems taken from the CUTE test problems.

The initial point x_0 is obtained from the CUTE collection. If there is a j-th component of x_0 such that $x_0^j \leq 0$, we redefine $x_0^j = 2$. From x_0, we

compute $z_0 = be$ where $b = \max\{0.1, \|\nabla f(x_0)\|\}$, and $y_0 = \|\nabla f(x_0)\| e_m$ with $e_m = (1, \ldots, 1)^T \in \mathbb{R}^m$. We set the initial perturbation parameter to be $\mu_0 = \frac{x_0^T z_0}{n}$. If the updated iterate $(x_{k+1}, z_{k+1}) \in N_W(\gamma\mu_k)$, with $\gamma = 0.8$, we calculate $\mu_{k+1} = \frac{N_W}{100}$.

When computing Δx_p and Δx_h, we let $\delta = 0.9$, $I_{\text{cg}} = \min\{n - m, 20\}$, and $\epsilon_1 = 10^{-6}$. We use a sparse Cholesky package to compute $(AA^T)^{-1}$, developed by Ng and Peyton, and obtained from LIPSOL [20]. In the computation of α_k^+, we let $\tau = 0.99995$, and to compute α_k, we set $s = [1 \;\; 0.8 \;\; 0.6 \;\; 0.5 \;\; 0.2]$ and $\beta = 10^{-4}$. In Algorithm 5.2, we let $c = 2$ as in [2, 4].

The trust region radius is modified in the usual way (See [16]) where the initial and maximum trust region radii are $\Delta_0 = 5$ and $\Delta_{\max} = 20$, respectively. If we have not accepted an inexact step after a maximum number of steps allowed $\max_{\text{tr}} = 5$ with $\eta = \frac{1}{4}$, we force the acceptance of the last step and set $\Delta = 2.5$. We set the algorithm's total number of iterations to $k_{max} = 300$ with a tolerance of $\epsilon = 10^{-7}$.

In Tables 3 and 4, we report the number of inexact or exact Newton steps computed for each of the 50 problems to converge. A dash in a column indicates that convergence was not achieved within k_{max} iterations. The last row in Table 4 states the total number of problems that converge for each algorithm.

The first three columns state the problem name, the dimension of x, and the number of equality constraints, respectively. The fourth column reports the results obtained by using Algorithm 5.3, that implements the proposed inexact Newton hybrid path-following algorithm. The next column shows the results obtained by using an exact Newton hybrid path-following algorithm, i.e. Algorithm 5.3 with a modified Step 2a in order to calculate exact Newton steps, and the following column modifies Algorithm 5.3 by omitting the linesearch technique within the trust region globalization strategy. Both of these results are reported in columns five and six, respectively.

In the previous section, we pointed out that the formation of the inexact Newton step is independent of the invertibility of the coefficient matrix in (2.5). As evidence, we refer the reader to the following problems of Tables 3 and 4: hs9, hs68, lsnnodoc, hs54, hs55, and robot. In these cases, Algorithm 5.3 achieves convergence and by using the exact Newton step, the algorithm fails to converge in less than $\max_{\text{iter}}$ iterations.

By examining Tables 3 and 4, there are cases where the exact Newton algorithm outperforms Algorithm 5.3. It may be the case that the inexact Newton step is not as good of an approximation to a solution of (2.5). From Proposition 4.6, we may calculate more accurate inexact steps by increasing $\max_{\text{tr}}$ in Algorithm 4.1.

In order to determine the effect of the linesearch within the trust region method on the Algorithm 5.3, we compare two implementations: Algorithm 5.3 with and without linesearch (excluding Step 2c).

TABLE 3. Numerical Results

Problem	n	m	Algorithm 5.3	Alg. 5.3 with Exact Newton Steps	Alg. 5.3 without linesearch
alsotame	2	1	14	22	17
extrasim	2	1	5	5	5
hs6	2	1	42	112	19
hs7	2	1	10	10	10
hs8	2	2	10	9	10
hs9	2	1	5	-	8
supersim	2	2	3	3	3
tame	2	1	3	3	3
tryb	2	1	12	14	11
aljazzaf	3	1	46	27	29
hs26	3	1	54	26	41
hs27	3	1	14	14	19
hs28	3	1	15	15	-
hs60	3	1	12	12	12
hs61	3	2	19	13	19
hs62	3	1	7	7	13
hs63	3	2	23	11	17
hong	4	1	8	8	13
hs39	4	2	11	23	12
hs40	4	3	5	5	5
hs42	4	2	12	11	12
hs68	4	2	89	-	47
hs69	4	2	15	15	21
lin	4	2	13	31	20
bt13	5	1	229	72	-
hs46	5	2	18	17	18
hs47	5	3	27	12	59
hs48	5	2	6	6	6
hs49	5	2	19	16	-
hs50	5	3	-	17	-
hs51	5	3	5	5	5
hs52	5	3	34	22	-
hs53	5	3	23	23	-
hs77	5	2	119	11	42
hs79	5	3	14	24	14

From Tables 3 and 4, the numerical results show that Algorithm 5.3 performs better than the version that omits the linesearch techniques. We conclude

TABLE 4. Numerical Results (Cont.)

Problem	n	m	Algorithm 5.3	Alg. 5.3 with Exact Newton Steps	Alg. 5.3 without linesearch
lsnnodoc	5	4	12	-	12
hs54	6	1	10	-	10
hs55	6	6	25	-	43
hs56	7	4	16	10	17
hs99	7	2	64	8	56
trigger	7	6	21	19	21
hs112	10	3	39	19	32
odfits	10	6	55	13	55
robot	14	2	7	-	8
hs119	16	8	86	49	-
fccu	19	8	46	22	35
degenlpa	20	15	74	61	-
degenlpb	20	15	55	56	-
himmelbk	24	14	107	45	-
optcntrl	32	20	20	140	143
TOTAL			49	44	40

that Algorithm 5.3 is more robust: 49 problems converge with it, and only 40 converge without.

6. **Conclusions**

We introduced an inexact Newton hybrid path-following algorithm for solving nonlinear programming problems. We make use of inexact Newton steps that allows us to relax the Newton assumptions associated to this problem, and obtain these steps more efficiently from a computational point of view. The hybrid path-following strategy incorporates a linesearch into the trust region method.

The algorithm was tested on a set of CUTE test problems obtaining promising results. We perform a numerical comparison of our proposed inexact Newton hybrid path-following algorithm versus the exact Newton hybrid path-following algorithm and inexact Newton trust region algorithm. The numerical results obtained show that our algorithm is more robust than the other two algorithms. Furthermore, our algorithm is able to solve degenerate problems. These preliminary results motivate a further research of the hybrid path-following strategy. Future work involves the addition of preconditioners within the conjugate gradient algorithm for solving large scale problems efficiently, and to prove global convergence theory.

Acknowledgments: The authors were supported in part by the U.S. Department of the Army Grant No. DAAD19-01-1-0741 and DAAD19-02-1-0243. The first and third authors are assistant and associate professors in the Department of Mathematical Sciences, University of Texas at El Paso, respectively. The second author earned a master degree under the direction of Drs. Argáez and Velázquez.

References

[1] M. Argáez, An inexact projected Newton method for large scale linear programming, Technical Report, Department of Mathematical Sciences, The University of Texas at El Paso, 2006.

[2] M. Argáez, Exact and inexact Newton linesearch interior-point algorithms for nonlinear programming problems, Ph.D Thesis, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 1997.

[3] M. Argáez, H. Klie, M. Ramé & R. A. Tapia, An interior-point Krylov-Orthogonal projection method for nonlinear programming, *Tec. Rep. 97–16*, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 1997.

[4] M. Argáez & R. A. Tapia, On the global convergence of a modified augmented Lagrangian linesearch interior-point Newton method for nonlinear programming, *J. Optim. Theory Appl.*, **114** (2002), 1–25.

[5] M. Argáez, R. A. Tapia & L. Velázquez, Numerical Comparisons of path-following strategies for a primal-dual Interior-Point method for nonlinear programming. *J. Optim. Theory Appl.*, **114** (2002) 4, 255–272.

[6] M. Argáez, L. Velázquez & M.C. Villalobos, On the development of a trust region interior-point method for large scale nonlinear programs, ACM Proc. in *Diversity of Comput.* Conf., (2003), 14–16.

[7] M. Argáez, O. Mendez & L. Velázquez, 2005. The notion of the quasicentral path for linear programming, *Technical Report*, Department of Mathematical Sciences, The University of Texas at El Paso, El Paso, Texas, 2006.

[8] R. Byrd, J.C. Gilbert & J. Nocedal, A trust region method based on interior point techniques for nonlinear programming, *Math. Program. A*, **89** (2000), 149–185.

[9] J.E. Dennis, Jr., & R.B. Schnabel, Numerical methods for unconstrained optimization and nonlinear equations, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia PA, 1996.

[10] M. El-Alem, A global convergence theory for a class of trust region algorithms for constrained optimization, Ph.D Thesis, Rice University, Houston, Texas, (1998).

[11] A.S. El-Bakry, R.A. Tapia, T. Tsuchiya & Y. Zhang, On the formulation of the primal-dual interior-point method for nonlinear programming, *J. Optim. Theory Appl.*, **89** (1996), 507–541.

[12] N.I.M. Gould, M. E. Hribar, & J. Nocedal, On the solution of equality constrained quadratic programming problems arising in optimization, *SIAM J. Sci. Comput.*, **23** (2001), 1375–1394.

[13] M.R. Hestenes, Multiplier and gradient methods, *J. Optim. Theory Appl.*, **4** (1969), 303–320.
[14] D. Kincaid, & W. Cheney, Numerical analysis: mathematics of scientific computing, 3rd ed., Brooks-Cole, Boston MA, (2002).
[15] D.G. Luenberger, Linear and nonlinear programming, 2nd ed., Addison-Wesley, Reading MA, (1984).
[16] J. Nocedal & S.J. Wright, Numerical optimization, Springer-Verlag, (1999).
[17] R. Sáenz, An inexact Newton interior-point algorithm for nonnegative constrained minimization, *Master Thesis*, Department of Mathematical Sciences, The University of Texas at El Paso, El Paso, Texas (2003).
[18] D.f. Shanno & R. J. Vanderbei, Interior-point methods for nonconvex nonlinear programming: Orderings and higher-order methods, *Math. Prog.*, **87** (2000), 303-316.
[19] P. Tseng & Y. Ye, On some interior-point algorithms for nonconvex quadratic optimization, *Math. Prog.* **93** (2002), 217–225.
[20] Y. Zhang, Solving large-scale linear programs by interior-point methods under the MATLAB environment, *Optimization Methods and Software*, **10** (1998), 1–31.

(Recibido en junio de 2006. Aceptado en noviembre de 2006)

Department of Mathematical Sciences
The University of Texas
El Paso, TX USA
e-mail: margaez@utep.edu