**Department of Mathematics Report NA 02-1,**
University of California, San Diego.

E. Michael Gertz[1] · Philip E. Gill[2]

# A PRIMAL-DUAL TRUST-REGION ALGORITHM FOR NONLINEAR OPTIMIZATION*

7 October 2002; Revised 25 June 2003

**Abstract.** This paper concerns general (nonconvex) nonlinear optimization when first and second derivatives of the objective and constraint functions are available. The proposed method is based on finding an approximate solution of a sequence of unconstrained subproblems parameterized by a scalar parameter. The objective function of each unconstrained subproblem is an augmented penalty-barrier function that involves both primal and dual variables. Each subproblem is solved using a second-derivative Newton-type method that employs a combined trust-region and line search strategy to ensure global convergence. It is shown that the trust-region step can be computed by factorizing a sequence of systems with diagonally-modified primal-dual structure, where the inertia of these systems can be determined without recourse to a special factorization method. This has the benefit that off-the-shelf linear system software can be used at all times, allowing the straightforward extension to large-scale problems.

---

**Key words.** nonlinear optimization – constrained minimization – primal-dual methods – interior methods – trust-region methods

## 1. Introduction

This paper concerns methods for solving the nonlinear programming problem:

$$\text{(NP)} \qquad \begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & f(x) \\ \text{subject to} & c_i(x) = 0, \quad i \in \mathcal{E}, \quad \text{and} \quad c_i(x) \geq 0, \quad i \in \mathcal{I}, \end{array}$$

where $c(x)$ is an $m$-vector of nonlinear constraint functions with $i$th component $c_i(x)$, $i = 1, \ldots, m$, and $\mathcal{E}$ and $\mathcal{I}$ are nonintersecting index sets. It is assumed throughout that $f$ and $c$ are twice-continuously differentiable.

Under certain conditions (see e.g. [19,26,60,64]), the first-order optimality conditions for (NP) may be perturbed by a positive parameter $\mu$ in such a way that the perturbed conditions define a differentiable *trajectory* or *central path* of solutions that passes through the solution as $\mu \to 0$. Primal-dual interior

---

* Also published as UW-Madison Computer Sciences Department, Optimization Technical Report 02-09, October 2002.

E. Michael Gertz: Computer Sciences Department, University of Wisconsin, Madison, WI 53706 e-mail: gertz@mcs.anl.gov

Philip E. Gill: Department of Mathematics, University of California, San Diego, La Jolla, CA 92093-0112 e-mail: pgill@ucsd.edu

*Mathematics Subject Classification (1991):* 49M37, 65F05, 65K05, 90C30

methods attempt to follow this path by applying a form of Newton's method to the perturbed system while maintaining strictly feasible estimates of the primal and dual variables. In this form, primal-dual methods have a two-level structure of inner and outer iterations: the inner iterations correspond to the iterations of Newton's method, and the outer iterations test for convergence and (if necessary) adjust the value of $\mu$.

Primal-dual methods exhibit excellent performance in the neighborhood of a solution. In particular, under the assumption of strict complementarity and a suitable constraint qualification, the inner iterations can be terminated in such a way that the combined sequence of inner iterates converges to $x^*$ at a Q-superlinear rate; see, e.g., [38,63,66,67]. Notwithstanding this excellent local convergence, the efficiency and reliability of primal-dual methods for general nonconvex problems are significantly affected by the following four related issues: (i) the method used to handle nonlinear equality constraints; (ii) the scheme used to ensure progress toward the solution; (iii) the formulation and solution of the linear system for the Newton step; and (iv) the treatment of nonconvexity.

Forsgren and Gill [23] introduced a primal-dual interior-point method based on minimizing an augmented penalty-barrier function. They further show how to globalize this method using a line search and an inertia-controlling factorization of the primal-dual matrix. In this paper we present a primal-dual method that is also based on minimizing the Forsgren-Gill penalty-barrier function. A combination trust-region technique, based on one of several algorithms given in Gertz [32], is used to handle nonconvexity and ensure convergence. The proposed method has several advantages:

- it may be implemented using off-the-shelf direct linear solvers;
- it requires only a modest number of factorizations;
- it has strong convergence theory; and
- it may be used to implement algorithms that maintain feasibility with respect to nonlinear inequality constraints.

In Section 2 we discuss how the trust-region subproblem may be solved by factoring matrices with the same structure as the primal-dual matrices. No special factorization need be used for the nonconvex case and hence the system can be solved by a range of efficient off-the-shelf direct solvers, or by methods tailored to particular applications.

In Section 3 we describe the combination trust-region line-search algorithm, which combines the best practical elements of line-search and trust-region algorithms without jeopardizing the strong convergence theory. The new method is particularly well-suited to an interior-point algorithm that maintains feasibility with respect to nonlinear inequality constraints. With conventional trust-region algorithms, the discovery that a trial step is infeasible typically requires a reduction in the trust-region radius and the solution of a new trust-region subproblem. Since a primal-dual system must be refactorized for each new subproblem, this can result in a significant cost per iteration. By contrast, the new method computes an improved primal-dual iterate every time a trust-region subproblem is solved.

Section 4 is devoted to the convergence theory of the new method. We show that under suitable conditions, the sequence of inner iterations converges to a point that satisfies the second-order necessary conditions for a minimizer of the Forsgren-Gill barrier-penalty function. Since this function is minimized at every point on the trajectory, the results provide the theoretical basis for an algorithm that converges to points satisfying the second-order necessary conditions for a minimizer of (NP) (see [33]).

In Section 5 we describe how to solve the primal-dual trust-region subproblem using a modified version of an algorithm due to Moré and Sorensen [45]. On average, this algorithm requires few matrix factorizations per trust-region solve and hence provides a practical way of implementing the algorithm given in this paper. In Section 6 we present numerical results obtained by applying a preliminary implementation of the primal-dual algorithm to a set of general nonlinear problems.

### *1.1. Related work*

We now summarize some related work on primal-dual interior-point algorithms for nonlinear programming. These methods are an active field of research, and we are only able to touch on some of the current work here. For a more complete survey, see Forsgren, Gill and Wright [26].

Some of the earliest primal-dual interior algorithms for general nonlinear problems were modeled on methods for convex programming. In the convex case the linearized primal-dual equations provide a direction for a line search on the two-norm of the residuals of the perturbed equations (see, e.g., [18,63]). In this situation, the normed residuals define a merit function whose value provides a measure of the proximity of an iterate to the solution.

*Barrier-SQP* methods are specifically designed for problems with a mixture of inequality and equality constraints and have their roots in general nonlinear programming. Barrier-SQP methods can be derived by interpreting the perturbed primal-dual system as the first-order optimality conditions for the minimization of a logarithmic barrier function subject to the original equality constraints (see Section 1.4). This equality constrained problem can then be solved using either a line-search or trust-region SQP method. Argaez and Tapia [2] propose a line-search barrier-SQP method in which the merit function includes an augmented Lagrangian term for the equality constraints. Gay, Overton and Wright [31] also use a combined augmented Lagrangian and logarithmic barrier merit function, but propose the use of a modified Cholesky factorization of the condensed Hessian (see Section 1.5) to handle nonconvex problems. The software package LOQO [4,53,57] is based an a barrier-SQP algorithm that uses a direct factorization to solve the full primal-dual Newton system. LOQO uses a line search to reduce a conventional penalty-barrier function (see Fiacco and McCormick [19]) and the Hessian of the Lagrangian is modified by a positive multiple of the identity if it appears that the problem is not locally convex.

Trust-region barrier-SQP methods are largely based on the Byrd-Omojokun algorithm [8,47], which uses different trust-regions to control the normal and tangential components of the SQP search direction (see, e.g., [9,10,14,59]). Yabe, Yamashita, Tanabe [65] use a trust-region approach based on a merit function that includes equalities as an $\ell_1$ penalty term. The code KNITRO is a trust-region barrier-SQP method that uses an exact $\ell_2$ merit function together with the conjugate-gradient method for finding approximate solutions of the normal and tangential trust-region subproblems [9,10,59]. Conn, Gould, Orban and Toint [12] propose a method for linear equality and nonlinear inequality constraints that uses the logarithmic barrier function in conjunction with a weighted two-norm trust region method (also, see Gonn, Gould and Toint [13]).

Methods that use the logarithmic barrier function to handle inequalities in the merit function must use a separate procedure to safeguard the dual variables because the dual variables do not appear in the merit function. For example, in Conn *et. al.* [12], after the primal variables have been updated by the trust-region subproblem, the dual variables are updated by a separate procedure that ensures progress towards the trajectory. By contrast, the line-search method of Forsgren and Gill [23] minimizes an augmented penalty-barrier function with respect to both the primal and dual variables. The formulation of an appropriate quadratic model based on this augmented function leads to a symmetric indefinite system that is identical to the conventional primal-dual system. Nonconvexity is handled using an inertia-controlling symmetric indefinite factorization (for more details, see Section 1.6 and Forsgren [22]).

A quite different procedure for forcing convergence is to use a *filter*, first proposed by Fletcher and Leyffer [21] in the context of SQP methods. A filter is a set of objective pairs that represents the Pareto frontier of best points for a bi-objective optimization problem. There are a number of different ways to define the filter in the primal-dual context, but a common choice is to use the norm of the Lagrangian gradient and the norm of the distance to the trajectory. At each inner iteration a line-search or trust-region algorithm is used to obtain a new entry for the filter. At each step, filter elements with objective pairs that are both worse than those at the new point are removed from the filter. For more details, see [3,56,58].

### *1.2. Notation*

The gradient of $f(x)$ is denoted by $\nabla f(x)$, and the $m \times n$ Jacobian of $c(x)$ is denoted by $J(x)$. Unless explicitly indicated, $\|\cdot\|$ denotes the vector two-norm or its subordinate matrix norm. The least eigenvalue of a symmetric matrix $A$ will be denoted by $\lambda_{\min}(A)$. Given a real symmetric matrix $A$, the inertia of $A$—denoted by $\mathrm{In}(A)$—is the associated integer triple $(a_+, a_-, a_0)$ indicating the number of positive, negative and zero eigenvalues of $A$. An infinite sequence of matrices $\{A_j\}$ is considered to be bounded if the sequence $\{\|A_j\|\}$ is bounded. Given vectors $x$ and $y$ of dimension $n_x$ and $n_y$, the $(n_x + n_y)$-vector of elements of $x$ augmented by elements of $y$ is denoted by $(x, y)$.

Following common usage in the interior-point literature, if a vector is denoted by a lower-case letter, the same upper-case letter denotes the diagonal matrix whose elements are those of the vector, so that $V = \text{diag}(v)$. Finally, $e$ denotes the vector of all ones whose dimension is determined by the context.

Let $\{\alpha_j\}_{j\geq 0}$ be a sequence of scalars, vectors or matrices and let $\{\beta_j\}_{j\geq 0}$ be a sequence of positive scalars. If there exists a positive constant $\gamma$ such that $\|\alpha_j\| \leq \gamma\beta_j$, we write $\alpha_j = O(\beta_j)$. If there exist positive constants $\gamma_1$ and $\gamma_2$ such that $\gamma_1\beta_j \leq \|\alpha_j\| \leq \gamma_2\beta_j$, we write $\alpha_j = \Theta(\beta_j)$.

### *1.3. Optimality conditions*

Subject to a constraint qualification (see, e.g., [20], [36], and [46, Chapter 12]) the first-order necessary optimality conditions for problem (NP) state that at an optimal solution $x^*$ there must exist an $m$-vector $y^*$ of Lagrange multipliers such that

$$\nabla f(x^*) - J(x^*)^T y^* = 0, \qquad \begin{aligned} c_i(x^*)y_i^* &= 0, \quad i \in \mathcal{I}, \qquad c_i(x^*) = 0, \quad i \in \mathcal{E}, \\ y_i^* &\geq 0, \quad i \in \mathcal{I}, \qquad c_i(x^*) \geq 0, \quad i \in \mathcal{I}. \end{aligned}$$

If we associate each constraint function $c_i(x)$ with a value $z_i = z_i(y_i)$ such that $z_i = 1$ for $i \in \mathcal{E}$, and $z_i = y_i$ for $i \in \mathcal{I}$, then we can write these conditions in the compact form $F^\infty(x^*, y^*) = 0$, and $y_i^* \geq 0$, $c_i(x^*) \geq 0$ for $i \in \mathcal{I}$, where $F^\infty(x, y)$ is the vector-valued function

$$F^\infty(x, y) = \begin{pmatrix} \nabla f(x) - J(x)^T y \\ C(x)z(y) \end{pmatrix}, \tag{1.1}$$

and $C(x) = \text{diag}\big(c_1(x),\, c_2(x),\, \ldots,\, c_m(x)\big)$.

### *1.4. Primal-dual methods*

Primal-dual methods can be interpreted as solving a sequence of nonlinear systems in which each condition $c_i(x)y_i = 0$, $i \in \mathcal{I}$ in (1.1) is perturbed as $c_i(x)y_i = \mu$ for some small positive $\mu$. This allows the perturbed equations to be solved using a form of Newton's method in which $x$ and $y$ are chosen to be interior for the inequalities $c_i(x) > 0$ and $y_i > 0$ for $i \in \mathcal{I}$ (see Section 1.5). The perturbed equations can also be interpreted as the first-order optimality conditions for the equality constrained problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) - \mu \sum_{i\in\mathcal{I}} \ln c_i(x), \quad \text{subject to} \quad c_i(x) = 0, \quad i \in \mathcal{E}, \tag{1.2}$$

with Lagrange multipliers for the inequalities being defined as $y_i = \mu/c_i(x)$ for $i \in \mathcal{I}$. Barrier-SQP methods exploit this interpretation by replacing the general mixed-constraint problem (NP) by a sequence of equality constraint problems in

which the inequalities are eliminated using a logarithmic barrier transformation (see, e.g., [2,9,12,14,31,57]).

The methods considered in this paper involve a perturbation of the *equality* constraints as well as the inequality constraints. The perturbation replaces all equalities $c_i(x) = 0$ by $c_i(x) + \mu y_i = 0$ (see, for instance, Gould [37] and Powell [49]). Here, for simplicity, the same perturbation is used for both equalities and inequalities, but in practice a different perturbation is used—see Section 6. If both equalities and inequalities are perturbed, we obtain a system of $n + m$ nonlinear equations $F^\mu(x, y) = 0$ such that

$$F^\mu(x, y) = \begin{pmatrix} \nabla f(x) - J(x)^T y \\ r(x, y) \end{pmatrix}, \tag{1.3}$$

where $r(x, y)$ is an $m$-vector with components $r_i(x, y) = c_i(x) + \mu y_i$ for $i \in \mathcal{E}$, and $r_i(x, y) = c_i(x)y_i - \mu$ for $i \in \mathcal{I}$. In this case, the identity $F^\mu(x, y) = 0$ can be interpreted as the first-order optimality conditions for the unconstrained penalty-barrier problem [19]:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \mathcal{B}^\mu(x) = f(x) - \mu \sum_{i\in\mathcal{I}} \ln c_i(x) + \frac{1}{2\mu} \sum_{i\in\mathcal{E}} c_i(x)^2. \tag{1.4}$$

Perturbing the equality constraints has the benefit of regularizing the problem, in the sense that as long as $\mu$ is nonzero, the equality-constraint Jacobian need not have full rank. (For a discussion of the role of regularization in linear and quadratic programming, see, e.g., [34,35]).

If the second-order sufficient conditions hold and the gradients of the active constraints are linearly independent, then for $\mu$ sufficiently small, a differentiable primal-dual trajectory of solutions $\big(x(\mu), y(\mu)\big)$ exists and converges to $\big(x^*, y^*\big)$ as $\mu \to 0^+$. Primal-dual interior methods attempt to follow this trajectory by finding an approximate solution of $F^\mu(x, y) = 0$ for a positive sequence of $\mu$-values that decreases monotonically to zero.

Algorithm 1.1 gives a generic primal-dual iteration for problem (NP). The quantity $F^{(k)}(x, y)$ is used to denote the residual function $F^\mu(x, y)$ for the particular value $\mu = \mu^{(k)}$. An inner iterate $\big(x, y\big)$ is considered to be an acceptable approximate zero of $F^\mu$ if $t\big(x, y, \mu\big) \leq \mu$, where $t\big(x, y, \mu\big)$ is a nonnegative function such that: (i) $\|F^\mu(x, y)\| \leq \mu^{(k)}$ implies $t\big(x, y, \mu\big) \leq \mu^{(k)}$; and (ii) as $\mu^{(k)} \to 0^+$, convergence of $\{\, t\big(x^{(k)}, y^{(k)}, \mu^{(k)}\big) \,\}$ to zero implies convergence of $\{\, F^\infty(x^{(k)}, y^{(k)}) \,\}$ to zero. The simple choice of $t\big(x, y, \mu\big) = \|F^\mu(x, y)\|_\infty$ is suitable for theoretical purposes, but other choices may be better for practical computation (see Section 6.2 for another definition). Note that the termination criterion is loose for early iterates, but becomes progressively tighter as the iterations proceed.

Algorithm 1.1. Generic Primal-Dual Iteration.
Specify parameters $\mu^{(0)}$, *tol* and $k_{\max}$;
Choose $x^{(0)}$, $y^{(0)}$;

$k \leftarrow 1$;
**repeat**
  *Inner iteration:* Find $(x^{(k)}, y^{(k)})$ so that
    $t\big(x^{(k)}, y^{(k)}, \mu^{(k)}\big) \leq \mu^{(k)}$ **or** $\|F^{\infty}(x^{(k)}, y^{(k)})\|_{\infty} \leq tol$;
  *converged* $\leftarrow \{\|F^{\infty}(x^{(k)}, y^{(k)})\| \leq tol\}$;
  **if not** *converged* **then**
    Choose $\mu^{(k+1)} < \mu^{(k)}$.
  **end if**
  $k \leftarrow k + 1$;
**until** *converged* **or** $k > k_{\max}$;

Ideally, one would choose a sequence $\{\mu^{(k)}\}$ that converges to zero at a superlinear rate. This was shown by Zhang, Tapia and Dennis [68] to be a necessary condition for superlinear convergence of the iterates $\big(x^{(k)}, y^{(k)}\big)$. In practice, however, a fast linear reduction of $\mu^{(k)}$ will serve just as well, because the cost of computing the final iterates is typically much less than the cost of computing the initial iterates. A suitable rule for choosing $\{\mu^{(k)}\}$ is discussed in Section 6. The method for finding an acceptable inner iterate $(x^{(k)}, y^{(k)})$ is given in Section 3.

The emphasis of this paper is on a provably convergent method for generating the inner iterates. Accordingly, we do not consider the affect of the choice of $\mu$ update on the global convergence of the outer iterations. We defer such issues to a later paper.

### *1.5. Definition of the inner iterations*

In this section we focus on the inner iteration. Since the discussion involves calculations associated with a fixed $\mu$, the superscript $k$ will be omitted for the remainder of the section. Let $v$ denote the $n+m$ vector of the combined unknowns $(x, y)$ at an interior point, i.e., a point such that $c_i(x) > 0$ and $y_i > 0$ for $i \in \mathcal{I}$. If $F^{\mu}(v)$ denotes the function $F^{\mu}(x, y)$, then a Newton direction $\Delta v = (\Delta x, \Delta y)$ is defined by the Newton equations $F^{\mu}(v)'\Delta v = -F^{\mu}(v)$. It is convenient to write these equations in terms of the vector $\pi^{\mu}(x)$ of *primal multipliers*, which has components:

$$\pi_i^{\mu}(x) = \begin{cases} -c_i(x)/\mu & \text{for } i \in \mathcal{E}, \\ \mu/c_i(x) & \text{for } i \in \mathcal{I}. \end{cases} \tag{1.5}$$

This definition implies that the Newton equations may be expressed as

$$\begin{pmatrix} H(x,y) & -J(x)^T \\ Z(y)J(x) & \Gamma \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - J(x)^T y \\ \Gamma\big(y - \pi^{\mu}(x)\big) \end{pmatrix}, \tag{1.6}$$

where $\Gamma = \operatorname{diag}(\gamma_1, \gamma_2, \ldots, \gamma_m)$, with $\gamma_i = c_i(x)$ for $i \in \mathcal{I}$ and $\gamma_i = \mu$ for $i \in \mathcal{E}$, and $H(x, y)$ denotes the Hessian of the Lagrangian with respect to $x$, i.e., the matrix $H(x, y) = \nabla^2 f(x) - \sum_{i=1}^{m} y_i \nabla^2 c_i(x)$.

As described above, primal-dual methods have a two-level structure of inner and outer iterations, with the inner iterations corresponding to the iterations of

Newton's method. The cost of a primal-dual iteration is dominated by the cost of solving the linear system (1.6), and effective sparse linear system software is the key to efficiency for large problems. Since $Z$ and $\Gamma$ are invertible at an interior point, one approach is to use block elimination to obtain a smaller "condensed" system with matrix $H+J^TW^{-1}J$, where $W = Z^{-1}\Gamma$. This system can be solved by either a direct or iterative method. At points near a trajectory of minimizers, $H + J^TW^{-1}J$ is positive semidefinite and off-the-shelf sparse Cholesky or preconditioned conjugate-gradient software can be used.

An alternative approach is to use a direct solver for the full $(n+m)\times(n+m)$ system. The efficiency of a direct sparse solver should be less dependent on $H + J^TW^{-1}J$ being sparse (see, e.g., [29,34]). In this situation it is customary to symmetrize the system so that a symmetric solver can be used. For example, an equivalent symmetric form of (1.6) is

$$\begin{pmatrix} H & J^T \\ J & -W \end{pmatrix} \begin{pmatrix} \Delta x \\ -\Delta y \end{pmatrix} = - \begin{pmatrix} \nabla f - J^T y \\ W(y-\pi) \end{pmatrix}. \tag{1.7}$$

(The dependencies on $x$, $y$ and $\mu$ have been suppressed for clarity.) This particular form of symmetrization gives an ill-conditioned system, but the ill-conditioning is benign as long as certain direct methods are used to factorize the matrix (for more details, see [25,48,61,62]).

To ensure global convergence it is necessary to use a merit function to force the early iterates towards the trajectory. In the convex case, a typical strategy is to impose additional conditions on the step length $\alpha$ that ensure a sufficient reduction of some norm of the residual of the nonlinear equations (1.3), i.e., $\|F^\mu(v+\alpha\Delta v)\| < \|F^\mu(v)\|$ (see, e.g., [18,63]). The choice of merit function for nonconvex problem is considerably more complicated. The merit function $\|F^\mu(v)\|$ is not suitable in this case because it encourages convergence to points satisfying the first-order conditions and does not provide a mechanism for using second-order information. Some merit functions exploit the fact that equations (1.6) are the optimality conditions for the subproblems generated by an SQP method for the problem (1.2). This provides the motivation for many algorithms based on standard line-search or trust-region SQP strategies (see, e.g., [2,9,10, 14,31]). These algorithms usually converge to a second-order point, although this can be guaranteed for only some of the methods.

In the nonconvex case the situation is complicated by the fact that the solution of (1.6) may not exist or may not be a descent direction for the merit function. In this case the trust-region or line-search strategy must define (either implicitly or explicitly) a search direction from a related positive semidefinite system of the form $\bar{H} + J^TW^{-1}J$.

### *1.6. The augmented barrier-penalty function*

We now describe the Forsgren-Gill augmented barrier-penalty function and summarize some of its properties. For a more complete description, we refer the reader to [23].

If both equality and inequality constraints are perturbed, progress toward a local minimizer can be obtained by regarding the primal-dual iterates as minimizing the sequence of barrier-penalty functions $\mathcal{B}^\mu(x)$, introduced in equation (1.4), for each value of $\mu$. However, if $\mathcal{B}^\mu(x)$ is used as a merit function, it does not provide a measure of progress in the dual variables simply because it does not have terms involving $y$. An alternative approach is based on the properties of the Forsgren-Gill augmented barrier-penalty function:

$$\mathcal{M}^\mu(x,y) = f(x) - \mu\sum_{i\in\mathcal{I}} \ln c_i(x) + \frac{1}{2\mu}\sum_{i\in\mathcal{E}} c_i(x)^2 \\ - \mu\sum_{i\in\mathcal{I}}\left(\ln\left(\frac{c_i(x)y_i}{\mu}\right) + \frac{\mu - c_i(x)y_i}{\mu}\right) + \frac{1}{2\mu}\sum_{i\in\mathcal{E}}\big(c_i(x)+\mu y_i\big)^2, \quad (1.8)$$

which is the penalty-barrier function $\mathcal{B}^\mu(x)$ augmented by a weighted proximity term that measures the distance of $(x,y)$ to the trajectory $\big(x(\mu),y(\mu)\big)$. A fundamental property of $\mathcal{M}^\mu(x,y)$ is that it is minimized with respect to both $x$ and $y$ at any point $(x(\mu),y(\mu))$ on the trajectory (see Lemma 4.1 below). This property suggests that a decrease in $\mathcal{M}^\mu(x,y)$ may be used as a measure of progress toward a minimizer of $\mathcal{B}^\mu(x)$. However, unlike $\mathcal{B}^\mu(x)$, the function $\mathcal{M}^\mu(x,y)$ also measures progress resulting from changes in the dual variables $y$.

Forsgren and Gill propose a method for computing $\big(x(\mu),y(\mu)\big)$ based on the unconstrained minimization of $\mathcal{M}^\mu(x,y)$. This method uses a direction $s$ in both the primal and dual variables that is defined in terms of the local quadratic model $\mathcal{Q}(s) = g^Ts + \frac{1}{2}s^TBs$, where

$$g = \begin{pmatrix} \nabla f - J^T(2\pi - y) \\ W(y-\pi) \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} H + 2J^TW^{-1}J & J^T \\ J & W \end{pmatrix}. \quad (1.9)$$

The vector $g$ is the gradient of the merit function $\nabla\mathcal{M}^\mu$ written in terms of $W$ and the primal multipliers $\pi$, introduced in (1.7). The matrix $B$ approximates the Hessian $\nabla^2\mathcal{M}^\mu$ in the sense that if $\big(x,y\big) = \big(x(\mu),y(\mu)\big)$ is a point on the trajectory, then $B = \nabla^2\mathcal{M}^\mu$ and the approximation is exact. It can be shown that $B$ is positive definite if and only if $H+J^TW^{-1}J$ is positive definite. It follows that if $H + J^TW^{-1}J$ is positive definite, then the solution of the symmetric positive-definite system $Bs = -g$ is the unique minimizer of $\mathcal{Q}$. The crucial feature of this choice of $B$ is that $s = (\Delta x, \Delta y)$ is also a solution of the primal-dual system (1.6). This implies that algorithms that solve the primal-dual equations are implicitly using an approximate Newton method to minimize $\mathcal{M}^\mu$.

These properties form the theoretical basis of a line-search method that uses the solution of the primal-dual system (1.6) as search direction. If $B$ is sufficiently positive definite, then the search direction is the unique solution of $Bs = -g$ (equivalent to the primal-dual system (1.6)). Otherwise, the search direction is a linear combination of two vectors that are by-products of the relevant factorization of (1.7). The first vector is the solution of a related positive-definite system $\bar{B}s = -g$, the second vector is a direction of negative curvature for the

quadratic model $\mathcal{Q}(s)$. Methods for obtaining these search directions through inertia-controlling factorizations are described in [23,24,27]. A potential drawback of these approaches is that the row and column interchanges needed by the inertia-controlling factorization interfere with the row and column ordering used to maintain sparsity in the factors. This gives factors that are generally less sparse than those obtained by off-the-shelf sparse-matrix software. Moreover, since a modified factorization algorithm must be used, it is not easy to exploit state-of-the-art software, or software developed for specific types of advanced architectures.

In the remainder of this paper we formulate and analyze a primal-dual method in which the augmented penalty-barrier function is minimized using a combined trust-region and line search strategy. The proposed method has the benefit that off-the-shelf linear system software can be used at all times, allowing the straightforward extension to large-scale problems.

## 2. Globalization using trust regions

In this section we focus on the inner iteration and describe a trust-region method that finds an approximate minimizer of the merit function $\mathcal{M}^\mu$ for a given $\mu = \mu_k$. As $\mu$ is fixed during this process, we will write $\mathcal{M} = \mathcal{M}^\mu$ and $\mathcal{B} = \mathcal{B}^\mu$ where appropriate. The iterates of the trust-region method will be denoted by $v_j = (x_j, y_j)$. Similarly, the subscript $j$ will be used to indicate quantities that have been evaluated at $v_j$ (e.g., $H_j = H(x_j, y_j)$). The $i$th element of a vector associated with iteration $j$ is denoted by the subscript $ij$ (e.g., $y_{ij}$ is the $i$th element of the dual vector $y_j$).

Consider the quadratic model $\mathcal{Q}_j(s) = g_j^T s + \frac{1}{2} s^T B_j s$, where $g_j$ and $B_j$ are the quantities (1.9) defined at $v_j = (x_j, y_j)$, i.e.,

$$g_j = \begin{pmatrix} \nabla f_j - J_j^T(2\pi_j - y_j) \\ W_j(y_j - \pi_j) \end{pmatrix}, \quad B_j = \begin{pmatrix} H_j + 2J_j^T W_j^{-1} J_j & J_j^T \\ J_j & W_j \end{pmatrix}.$$

If $B_j$ is positive definite, the model $\mathcal{Q}_j(s)$ is minimized by the vector $s_j$ satisfying $B_j s = -g_j$. As discussed in Section 1.4, the system $B_j s = -g_j$ is equivalent to the primal-dual Newton system (1.6) and its symmetrized form

$$\begin{pmatrix} H_j & J_j^T \\ J_j & -W_j \end{pmatrix} \begin{pmatrix} \Delta x_j \\ -\Delta y_j \end{pmatrix} = - \begin{pmatrix} \nabla f_j - J_j^T y_j \\ W_j(y_j - \pi_j) \end{pmatrix}. \tag{2.1}$$

Trust-region methods minimize a quadratic model of the objective function subject to a restriction on the length of the step. The algorithm is based on finding an approximate solution of the trust-region subproblem

$$\underset{s \in \mathbb{R}^{n+m}}{\text{minimize}} \;\; \mathcal{Q}_j(s) \quad \text{subject to} \;\; \|s\|_{T_j} \leq \delta_j, \tag{2.2}$$

where $\|\cdot\|_{T_j}$ denotes the elliptic norm $\|s\|_{T_j} = (s^T T_j s)^{1/2}$ and $\delta_j$ is the trust-region radius. The matrix $T_j$ is a block-diagonal matrix of the form $T_j =$

diag$(M_j, N_j)$, where $M_j$ and $N_j$ are $n \times n$ and $m \times m$ symmetric positive-definite matrices defined below.

Several successful algorithms for solving the trust-region subproblem are based on the following theorem. (For a proof, see, e.g., Gay [30] and Sorensen [54].)

**Theorem 2.1.** *A vector $s_j$ is a global solution of the trust-region subproblem (2.2) if and only if $\|s_j\|_{T_j} \le \delta_j$ and there exists a nonnegative $\sigma_j$ such that*

$$(B_j + \sigma_j T_j)s_j = -g_j \quad \text{and} \quad \sigma_j\big(\delta_j - \|s_j\|_{T_j}\big) = 0, \tag{2.3}$$

*with $B_j + \sigma_j T_j$ positive semidefinite. For any global minimizer $s_j$, the value of $\sigma_j$ is unique. Furthermore, if a global minimum is achieved for more than one $s_j$, then $\sigma_j$ is independent of $s_j$. If $B_j + \sigma_j T_j$ is positive definite, then the global solution $s_j$ is also unique.* □

Moré and Sorensen [45] define efficient methods for finding an approximate solution of the trust-region subproblem (2.2). Their technique finds an optimal $\sigma_j$ by computing the Cholesky factorization of $B_j + \sigma T_j$ for various values of $\sigma$. In Section 3, we describe a modified version of the Moré-Sorensen algorithm that repeatedly solves systems of the form $B_j(\sigma)\Delta v = -g_j$, where

$$B_j(\sigma) \triangleq B_j + \sigma T_j = \begin{pmatrix} H_j + 2J_j^T W_j^{-1} J_j + \sigma M_j & J_j^T \\ J_j & W_j + \sigma N_j \end{pmatrix}. \tag{2.4}$$

Instead of using the Cholesky algorithm to factorize $B_j(\sigma)$ directly, we show that with appropriate choices of $M_j$ and $N_j$, the solution of $B_j(\sigma)\Delta v = -g_j$ can be obtained by solving an equivalent system that has the same "primal-dual" structure as (1.7). This equivalence can be established by multiplying both sides of (2.3) by the nonsingular matrix

$$\begin{pmatrix} I & -2J_j^T W_j^{-1} \\ 0 & I \end{pmatrix}$$

and symmetrizing. This yields the system

$$\begin{pmatrix} H_j + \sigma M_j & J_j^T \\ J_j & -D_j(\sigma) \end{pmatrix} \begin{pmatrix} \Delta x_j \\ -\Delta u_j \end{pmatrix} = - \begin{pmatrix} \nabla f_j - J_j^T y_j \\ W_j(y_j - \pi_j) \end{pmatrix}, \tag{2.5}$$

where $\Delta u_j = (I + 2\sigma W_j^{-1} N_j)\Delta y_j$ and $D_j(\sigma) = (W_j + \sigma N_j)(I + 2\sigma W_j^{-1} N_j)^{-1}$. If $N_j$ is chosen to be diagonal, then $D_j(\sigma)$ is also diagonal, with

$$d_{ij}(\sigma) = w_{ij}\big(w_{ij} + \sigma n_{ij}\big)/\big(w_{ij} + 2\sigma n_{ij}\big),$$

where $w_{ij}$, $n_{ij}$ and $d_{ij}(\sigma)$ denote the $i$th diagonal elements of $W_j$, $N_j$ and $D_j(\sigma)$ respectively. In particular, if we choose $N_j = W_j$, then $D_j(\sigma)$ has the simple form

$$d_{ij}(\sigma) = w_{ij}\big(1 + \sigma\big)/\big(1 + 2\sigma\big).$$

The positive-definite diagonal $M_j$ is arbitrary apart from the requirement that the sequences $\{M_j\}$ and $\{M_j^{-1}\}$ be bounded. In practice $M_j = I$ is an acceptable choice. When $M_j$ and $N_j$ are diagonal, the matrices of the trust-region subproblem have the same sparsity pattern as the primal-dual matrix (2.1).

Let the matrix of equation (2.5) be denoted by $K_j(\sigma)$, i.e.,

$$K_j(\sigma) = \begin{pmatrix} H_j + \sigma M_j & J_j^T \\ J_j & -D_j(\sigma) \end{pmatrix}. \tag{2.6}$$

A solution of the trust-region subproblem (2.2) is obtained by repeatedly factoring $K_j(\sigma)$ and solving the system (2.5) for different values of $\sigma$. The need to satisfy the conditions of Theorem 2.1 implies that we are interested in values of $\sigma$ for which $B_j(\sigma)$ is positive semidefinite. The following lemma shows that the inertia of $B_j(\sigma)$ (which is not factored explicitly) can be deduced from the inertia of $K_j(\sigma)$.

**Lemma 2.1.** *The inertias of the matrices $B_j(\sigma)$ and $K_j(\sigma)$ satisfy the identities:*

$$\begin{aligned} \text{In}\big(B_j(\sigma)\big) &= \text{In}\big(H_j + \sigma M_j + J_j^T D_j(\sigma)^{-1} J_j\big) + \big(m, 0, 0\big) \\ \text{In}\big(K_j(\sigma)\big) &= \text{In}\big(H_j + \sigma M_j + J_j^T D_j(\sigma)^{-1} J_j\big) + \big(0, m, 0\big). \end{aligned}$$

*Proof.* See Forsgren and Gill [23].

Given a symmetric indefinite factorization $P^T K_j(\sigma) P = LBL^T$, the number of negative eigenvalues of $K_j(\sigma)$ is the number of $2 \times 2$ blocks and negative $1 \times 1$ blocks of $B$ (see, e.g., Bunch and Parlett [7], Bunch and Kaufmann [6]). Lemma 2.1 implies that if $K_j(\sigma)$ has more than $m$ negative eigenvalues, then $B_j(\sigma)$ is not positive semidefinite and $\sigma$ must be increased.

The matrix associated with the system (2.5) must be refactorized for each new value of $\sigma$ and it is important that $\Delta v_j$ be computed with as few factorizations as possible. The combined line-search/trust-region method described in Section 3 requires significantly fewer factorizations than conventional methods without sacrificing the strong convergence results (for more details, see Gertz [32]).

## 3. The trust-region algorithm

Trust-region algorithms have excellent theoretical properties and work well in practice. Moreover, a trust-region approach allows the use of standard off-the-shelf linear solvers, even when the problem is nonconvex. However, trust-region methods may need the solution of more than one linear system to compute an inner iterate. Because finding a solution of this linear system can constitute a significant portion of computing time, performing a line search along the trust-region step can reduce the number of systems to be solved and hence significantly reduce the solution time.

In the case of primal-dual interior-point methods, there is an even more compelling reason to use a line search in the trust-region subproblem. In this case, the function being minimized includes a barrier function that is undefined outside the feasible region. It is not uncommon for the trust-region step to be rejected simply because it generates an iterate that is infeasible for the inequality constraints. In terms of additional matrix factorizations, it can be very expensive to search for a feasible step by repeatedly solving the trust-region subproblem for smaller values of the trust-region radius. If any of the inequality constraints are nonlinear, this search can require several iterations. Thus, it seems appropriate to use a line search to find a step that remains feasible.

Gertz [32] shows that a backtracking line search may be added to a trust-region algorithm without sacrificing the strong second-order convergence properties. A simple backtracking line search is considered in this paper, but more advanced line search techniques are also possible (see Gertz [32] for an algorithm based on the so-called strong Wolfe conditions).

Given a trust-region solution $s_j$, the backtracking line search finds a step length $\alpha_j$ that satisfies the Armijo-style condition

$$\mathcal{M}(v_j) - \mathcal{M}(v_j + \alpha_j s_j) \geq -\eta_1 q_j(\alpha_j s_j), \tag{3.1}$$

where $\eta_1 \in (0, \frac{1}{2})$ and $q_j(s)$ denotes the quadratic model

$$q_j(s) = g_j^T s + \tfrac{1}{2} \min\left(0, s^T B_j s\right). \tag{3.2}$$

If we define the merit function $\mathcal{M}(v)$ to be positively infinite outside the feasible region, only strictly feasible steps can satisfy this condition.

If $s_j^T B_j s_j \leq 0$ the Armijo-style condition (3.1) is equivalent to the condition

$$\mathcal{M}(v_j) - \mathcal{M}(v_j + \alpha_j s_j) \geq -\eta_1 \mathcal{Q}_j(\alpha_j s_j), \tag{3.3}$$

where $\mathcal{Q}_j(s) = g_j^T s + \frac{1}{2} s^T B_j s$. In this case, the line search success criterion is the usual trust-region success criterion $(\mathcal{M}(v_j + s_j) - \mathcal{M}(v_j))/\mathcal{Q}_j(s_j) \geq \eta_1$.

If $s_j^T B_j s_j > 0$, the requirement (3.1) is more stringent than (3.3) and is likely to have some advantage, particularly when $B_j$ is not a very accurate approximation to $\nabla^2 \mathcal{M}(v_j)$. Furthermore, it can be shown that if $\eta_1 < \frac{1}{2}$, then condition (3.1) does not adversely affect the asymptotic rate of convergence (see Dennis and Moré [15]).

Another modification of the conventional trust-region algorithm is to reset $y_{j+1}$ with the primal multipliers $\pi_{j+1}$ if the line search reduces the step. In particular, if $\alpha_j < 1$, then $v_{j+1}$ is redefined to be $(x_{j+1}, y_{j+1})$, where

$$x_{j+1} = x_j + \alpha_j \Delta x_j \quad \text{and} \quad y_{j+1} = \pi(x_{j+1}). \tag{3.4}$$

The reset is added for efficiency, and does not effect the theoretical properties of the algorithm. If $\alpha_j < 1$ the full trust-region step does not decrease $\mathcal{M}(v)$ sufficiently, which may indicate that $\mathcal{Q}_j(s)$ is not a good model of $\mathcal{M}(v_j + s) - \mathcal{M}(v_j)$. Resetting the multipliers cannot increase the merit function (i.e., $\mathcal{M}(v_{j+1}) \leq \mathcal{M}(v_j + \alpha_j s_j)$) and has the benefit of redefining $B_{j+1}$ as the merit

function Hessian $\nabla^2\mathcal{M}(v_{j+1})$. In effect, the Hessian in the trust-region model is replaced by the Hessian of the merit function when the reduction in the merit function is not sufficiently close to the predicted reduction. This can be important when the iterates $v_j$ approach a boundary of the feasible region, where the $\pi(x)$ changes very rapidly.

The combined trust-region and backtracking line-search algorithm is defined as follows (more details on the convergence criterion is given in Section 6.1).

Algorithm 3.1. Armijo Trust-Region Algorithm.
Specify constants $0 < \eta_1 < \eta_2 < \frac{1}{2}$, $0 < \gamma_2 < 1 < \gamma_3$ and $1 \leq \nu < 1/\gamma_2$;
$j \leftarrow 1$; $\delta_j \leftarrow 1$; $v_j \leftarrow v_0$;
**while not** *converged* **do**
  Compute $s_j = (\Delta x_j, \Delta y_j)$, an approximate solution to the subproblem (2.2);
  $\rho_j \leftarrow (\mathcal{M}(v_j + s_j) - \mathcal{M}(v_j))/q_j(s_j)$;
  **if** $\rho_j \geq \eta_1$ **then**
    *Successful iteration:* $v_{j+1} \leftarrow v_j + s_j$;
    **if** $\rho_j \geq \eta_2$ **then** Set $\delta_{j+1} \in \big[\delta_j, \max(\delta_j, \gamma_3\|s_j\|_{T_j})\big]$ **else** $\delta_{j+1} \leftarrow \delta_j$;
  **else**
    Find the smallest positive integer $\ell$ such that $\alpha_j = \gamma_2^\ell$ satisfies (3.1);
    $x_{j+1} \leftarrow x_j + \alpha_j \Delta x_j$;
    *Reset multipliers:* $y_{j+1} = \pi(x_{j+1})$; $v_{j+1} = (x_{j+1}, y_{j+1})$;
    Choose $\delta_{j+1} \in \big[\|\alpha_j s_j\|_{T_j}, \nu\|\alpha_j s_j\|_{T_j}\big]$;
  **end**
  $j \leftarrow j + 1$;
**end**

The method for solving the trust-region subproblem provides a step $s_j$ satisfying a positive semidefinite system of the form $(B_j + \sigma_j T)s_j = -g_j$ (for details, see Section 5). This implies that the trust-region step satisfies $g_j^T s_j \leq 0$ and that a suitable step $\alpha_j$ exists such that (3.1) is satisfied (for a proof, see Moré and Sorensen [44]).

The constant $\nu$ is used to reflect that fact that when the optimal step lies on the boundary of the trust region, $s_j$ can only be computed to within some tolerance of $\delta_j$. Typically, $s_j$ will satisfy $\|s_j\|_{T_j} \geq (1/\nu)\delta_j$ or $s_j = -B_j^{-1}g_j$. In the algorithm of Section 5, the rule for choosing $\delta_{j+1} \in [\|\alpha_j s_j\|_{T_j}, \nu\|\alpha_j s_j\|_{T_j}]$ is

$$\delta_{j+1} = \begin{cases} \|\alpha_j s_j\|_{T_j} & \text{if } s_j = -B_j^{-1}g_j, \\ \alpha_j \delta_j & \text{otherwise.} \end{cases}$$

Algorithm 3.1 is designed to make progress at *every* iteration—not only for the successful iterations in which the trust-region step gives an improvement consistent with the quadratic model. If $\alpha_j < 1$, the resulting step $\alpha_j s_j$ is unlikely to minimize $\mathcal{Q}_j(s)$ subject to the trust-region constraint, but the condition (3.1) serves to ensure that the new iterate, even if not optimal, makes significant progress toward a minimizer. The idea is that fewer trust-region subproblems need be solved, with a corresponding reduction in the number of factorizations of the primal-dual matrix.

For efficiency, it is important that the trust-region subproblem (2.2) is not solved exactly. Given a fixed tolerance $\tau$ $(0 < \tau < 1)$, an approximate solution need only satisfy the conditions

$$\mathcal{Q}_j(s_j) \leq \tau \mathcal{Q}_j(s_j^c) \quad \text{and} \quad \|s_j\|_{T_j} \leq \delta_j, \tag{3.5}$$

where $s_j^c$ is the scaled Cauchy point, which is defined as the solution of the problem $\min_{s,\beta}\{\, \mathcal{Q}_j(s) : T_j s = \beta g_j,\ \|s\|_{T_j} \leq \delta_j \,\}$. The method described in Section 5 finds a step $s_j$ that satisfies the sufficient decrease conditions

$$\mathcal{Q}_j(s_j) \leq \tau \mathcal{Q}_j^* \quad \text{and} \quad \|s_j\|_{T_j} \leq \delta_j, \tag{3.6}$$

where $\mathcal{Q}_j^*$ is the unique minimum of $\mathcal{Q}_j(s)$ on $\{\, s : \|s\|_{T_j} \leq \delta_j \,\}$. The required conditions (3.5) on $s_j$ then follow from the inequalities $\mathcal{Q}_j(s_j) \leq \tau \mathcal{Q}_j^* \leq \tau \mathcal{Q}_j(s_j^c)$.

## 4. Theoretical discussion

Some care is needed when applying standard convergence results from unconstrained optimization to Algorithm 3.1. The merit function and its derivatives include barrier terms that are not uniformly continuous in the feasible region, even when the feasible region is compact. In this section the standard unconstrained convergence theory is adapted to the barrier case.

First, we investigate the properties of the inequality-constraint proximity term in the merit function.

**Lemma 4.1.** *The function $\psi(\xi) = -\mu \ln\big(\xi/\mu\big) - \mu + \xi$, defined for $\xi > 0$, is strictly convex and nonnegative. The minimizer of $\psi(\xi)$ occurs at $\xi = \mu$ with minimum value $\psi(\mu) = 0$. Furthermore, $\lim_{\xi \to 0^+} \psi(\xi) = \infty$ and $\lim_{\xi \to \infty} \psi(\xi) = \infty$.*

*Proof.* Differentiating $\psi(\xi)$ twice gives $\psi'(\xi) = -\mu/\xi + 1$ and $\psi''(\xi) = \mu/\xi^2$. It follows that $\psi'(\mu) = 0$ and $\psi''(\xi) > 0$ for all $\xi > 0$. These conditions imply that $\psi(\xi)$ is strictly convex and has a unique minimizer at $\xi = \mu$. Because the minimum value $\psi(\mu) = 0$, the function must be nonnegative. □

**Lemma 4.2.** *Assume that $\{c_i(x_j)\}$ is bounded above for all $i \in \mathcal{I}$, and that $\{f(x_j)\}$ is bounded below. Then, for all $i \in \mathcal{I}$:*

(a) *$\{c_i(x_j)\}$ is bounded away from zero;*
(b) *$\{c_i(x_j) y_{ij}\}$ is bounded away from zero and bounded above;*
(c) *$\{y_{ij}\}$ is bounded above; and*
(d) *$c_i(x_j)/y_{ij} = \Theta\big(c_i(x_j)^2\big)$.*

*Proof.* Let $f(x_j) \geq \bar{f}$ for all $j$. Omitting nonnegative quantities and bounding $f(x_j)$ by $\bar{f}$ in the definition (1.8) of $\mathcal{M}$ yields:

$$\mathcal{M}(x_0, y_0) > \mathcal{M}(x_j, y_j) \geq \bar{f} - \mu \sum_{i \in \mathcal{I}} \ln c_i(x_j).$$

Under the assumption that $\{c_i(x_j)\}$ is bounded above, this inequality implies that $\{c_i(x_j)\}$ is bounded away from zero.

Similarly, the definition of $\mathcal{M}$ and the properties of $\psi$ yield the inequality

$$\mathcal{M}(x_0, y_0) > \mathcal{M}(x_j, y_j) \geq \bar{f} + \psi\big(c_i(x_j)y_{ij}\big) - \mu \sum_{i \in \mathcal{I}} \ln c_i(x_j) \quad \text{for} \quad i \in \mathcal{I}.$$

Lemma 4.1 and the bound on $\{c_i(x_j)\}$ imply the bound on $c_i(x_j)y_{ij}$. The upper bound (c) on $y_{ij}$ follows immediately from the bounds (a) and (b).

It follows from part (b) that $c_i(x_j)y_{ij}$ is bounded above and below. Some simple rearrangement yields the result (d). □

**Corollary 4.1.** *If $\{f(x_j)\}$ is bounded below and $\{c_i(x_j)\}$ is bounded above for all $i \in \mathcal{I}$, then the sequence $\{W_j^{-1/2}\}$ is bounded.*

*Proof.* For $i \in \mathcal{E}$ the diagonals of $W_j^{-1/2}$ are constant. For $i \in \mathcal{I}$, we have

$$1/w_{ij} = y_{ij}/c_i(x_j) = \Theta\big(1/c_i(x_j)^2\big), \tag{4.1}$$

and since $\{c_i(x_j)\}$ is bounded away from zero, the sequence $\{1/\sqrt{w_{ij}}\}$ is bounded above. □

Given the definitions of $B_j$ and $W_j$:

$$B_j = \begin{pmatrix} H_j + 2J_j^T W_j^{-1} J_j & J_j^T \\ J_j & W_j \end{pmatrix}, \quad \text{with} \quad w_{ij} = \begin{cases} \mu & \text{for } i \in \mathcal{E}, \\ c_i(x_j)/y_{ij} & \text{for } i \in \mathcal{I}. \end{cases}$$

Lemma 4.2 and Corollary 4.1 imply that $\|W_j^{-1/2}\|$ is bounded, but that $\|W_j\|$, and hence $\|B_j\|$, may grow without bound. This observation leads us to consider the transformed matrix

$$T_j^{-\frac{1}{2}} B_j T_j^{-\frac{1}{2}} = \begin{pmatrix} M_j^{-\frac{1}{2}}(H_j + 2J_j^T W_j^{-1} J_j)M_j^{-\frac{1}{2}} & M_j^{-\frac{1}{2}} J_j^T W_j^{-\frac{1}{2}} \\ W_j^{-\frac{1}{2}} J_j M_j^{-\frac{1}{2}} & I \end{pmatrix}, \tag{4.2}$$

which can remain bounded when $\|B_j\|$ does not. At each point $v = (x, y)$, consider the transformed quantities

$$\hat{g}(v) = T^{-\frac{1}{2}} g \quad \text{and} \quad \widehat{B}(v) = T^{-\frac{1}{2}} B T^{-\frac{1}{2}}. \tag{4.3}$$

Practical methods for solving the trust-region subproblem do not perform this transformation, but it is simpler to work with the rescaled matrix (4.2) when discussing the theoretical properties of the algorithm. With this notation, the trust-region subproblem (2.2) may be written in the form

$$\underset{\hat{s} \in \mathbb{R}^{n+m}}{\text{minimize}} \;\; \hat{g}_j^T \hat{s} + \tfrac{1}{2} \hat{s}^T \widehat{B}_j \hat{s} \quad \text{subject to} \quad \|\hat{s}\| \leq \delta_j, \tag{4.4}$$

where $\hat{s} = T_j^{1/2} s$. It will be shown that under suitable conditions, $\hat{g}_j$ converges to zero. This immediately leads to the question of what implications the convergence of $\hat{g}_j$ has to the convergence of $g_j$, or for that matter, to the convergence

of $F^{(k)}$. The relationship turns out to be nontrivial, but suggests that the convergence criterion requiring $\|F^{(k)}\|$ to be less than some tolerance is a reasonable one.

At issue is the convergence of the elements of $W_j(y_j - \pi_j)$ associated with the inequality constraints. These elements are related to the corresponding elements of $F^{(k)}$ through the relations

$$r_{ij} = c_i(x_j)y_{ij} - \mu = y_{ij}w_{ij}(y_{ij} - \pi_{ij}) \quad \text{for} \quad i \in \mathcal{I}. \tag{4.5}$$

In general, $\{y_{ij}\}$ is not bounded away from zero, which implies that the associated element of $g_j = \nabla\mathcal{M}$ can be arbitrarily larger than the corresponding element of $F^{(k)}$. Fortunately, as the following theorem shows, the situation is much improved when we work with $\hat{g}_j$ instead of $g_j$.

**Theorem 4.1.** *Assume that the sequences $\{M_j\}$ and $\{M_j^{-1}\}$ are bounded. If $\{f(x_j)\}$ is bounded below, $\{c_i(x_j)\}$ is bounded above for all $i \in \mathcal{I}$, and $J_j^T(y - \pi_j) = O\big(\|y_j - \pi_j\|\big)$, then $\hat{g}_j = \Theta\big(\|F^\mu(x_j, y_j)\|\big)$.*

*Proof.* From the definition of $\hat{g}$ (4.3) we have

$$\hat{g}_j = \begin{pmatrix} M_j^{-\frac{1}{2}}(\nabla f_j - J_j^T(2\pi_j - y_j)) \\ W_j^{\frac{1}{2}}(y_j - \pi_j) \end{pmatrix}.$$

The termination criterion involves the norm of the vector

$$F^\mu(x, y) = \begin{pmatrix} \nabla f(x) - J(x)^T y \\ r(x, y) \end{pmatrix}, \quad \text{where} \quad r_i(x, y) = \begin{cases} c_i + \mu y_i & \text{for } i \in \mathcal{E}, \\ c_i y_i - \mu & \text{for } i \in \mathcal{I}. \end{cases}$$

First, consider the vector $r(x, y)$ of last $m$ components of $F^\mu(x, y)$. It will be shown that

$$W_j^{\frac{1}{2}}(y_j - \pi_j) = \Theta\big(\|r(x_j, y_j)\|\big). \tag{4.6}$$

For $i \in \mathcal{E}$, the expression $\sqrt{w_{ij}}\,(y_{ij} - \pi_{ij}) = (c_i(x_j) + \mu y_{ij})\,/\sqrt{\mu}$ derived from the definition of $w_{ij}$ and $\pi_{ij}$ implies that the bound (4.6) holds for these components. Similarly, for $i \in \mathcal{I}$, we have

$$\sqrt{w_{ij}}\,(y_{ij} - \pi_{ij}) = \frac{c_i(x_j)y_{ij} - \mu}{\sqrt{c_i(x_j)y_{ij}}}.$$

In this case, the upper and lower bounds on $\{c_i(x_j)y_{ij}\}$ established in part (ii) of Lemma 4.2 imply that (4.6) holds.

Next, consider the first $n$ components of $F^\mu(x, y)$. If the sequences $\{M_j\}$ and $\{M_j^{-1}\}$ are bounded then

$$M_j^{-\frac{1}{2}}(\nabla f_j - J_j^T(2\pi_j - y_j)) = \Theta\big(\|\nabla f_j - J_j^T(2\pi_j - y_j)\|\big).$$

The term in parenthesis on the left-hand side can be expanded as

$$\nabla f_j - J_j^T(2\pi_j - y_j) = \nabla f_j - J_j^T y_j + 2J_j^T(y_j - \pi_j). \tag{4.7}$$

If it can be shown that $y_{ij}-\pi_{ij} = O\big(|F^{\mu}_{n+i}(x_j,y_j)|\big)$ and $y_{ij}-\pi_{ij} = O\big(|\hat{g}_{n+i}(v_j)|\big)$, then the assumption that $J_j^T(y_j-\pi_j) = O\big(\|y_j-\pi_j\|\big)$ and the relations (4.6) and (4.7) imply the result. The relations $y_{ij}-\pi_{ij} = O\big(|F^{\mu}_{n+i}(x_j,y_j)|\big)$ and $y_{ij}-\pi_{ij} = O\big(|\hat{g}_{n+i}(v_j)|\big)$ may be verified by substituting the definition of $\pi_{ij}$,

$$y_{ij} - \pi_{ij} = \begin{cases} (c_i(x_j) + \mu y_{ij})/\mu & \text{for } \; i \in \mathcal{E}, \\ (c_i(x_j)y_{ij} - \mu)/c_i(x_j) & \text{for } \; i \in \mathcal{I}. \end{cases} \tag{4.8}$$

For $i \in \mathcal{I}$, $c_i(x_j)$ is bounded away from zero and $y_{ij} - \pi_{ij} = O\big(|F^{\mu}_{n+i}(x_j,y_j)|\big)$. On the other hand, for $i \in \mathcal{E}$, $w_{ij}$ is constant and so $y_{ij} - \pi_{ij} = O\big(|\sqrt{w_{ij}}(y_{ij} - \pi_{ij})|\big)$. Furthermore, as we remarked earlier, equation (4.1) implies that $1/\sqrt{w_{ij}}$ is bounded, so $y_{ij} - \pi_{ij} = O\big(|\sqrt{w_{ij}}(y_{ij} - \pi_{ij})|\big)$, for $i \in \mathcal{I}$ as well, and thus $y_{ij} - \pi_{ij} = O\big(|\hat{g}_{n+i}(v_j)|\big)$. □

The assumption that $J_j^T(y_j - \pi_j) = O\big(\|y_j - \pi_j\|\big)$ certainly holds if $\{J_j\}$ is bounded. However, the assumption that $J_j^T(y_j - \pi_j) = O\big(\|y_j - \pi_j\|\big)$ is significantly weaker than the assumption that $\{J_j\}$ is bounded.

### *4.1. First-order analysis*

The convergence of Algorithm 3.1 is established using proofs similar to those given by Gertz [32] for the general unconstrained problem. However, the analysis is complicated by the need to occasionally reset the multipliers using (3.4). The possibility of a reset implies that the property $\|v_{j+1} - v_j\|_{T_j} \leq \delta_j$ may not hold at every step. What does remain true, however, is that $\|x_{j+1} - x_j\|_{M_j} \leq \delta_j$. This is sufficient to force convergence because the reset multipliers are a continuous function of $x$.

The proposed algorithm conforms to a general framework for proving the convergence of trust-region algorithms that was first presented in the early proofs of Powell [50,51]. To fit in this framework, an algorithm must have two properties:

($\mathbf{P}_1$) If $\{\mathcal{M}(v_j)\}$ is bounded below and $\{\|\hat{g}_j\|\}$ is bounded away from zero, then $\delta_j \to 0$ and $\{v_j\}$ converges.
($\mathbf{P}_2$) If $\{\|\hat{g}_j\|\}$ is bounded away from zero and $\{v_j\}$ converges, then $\delta_j \not\to 0$.

It follows immediately that for any algorithm that satisfies ($\mathbf{P}_1$) and ($\mathbf{P}_2$), either $\{\mathcal{M}(v_j)\}$ is unbounded below or $\liminf_{j\to\infty} \|\hat{g}_j\| = 0$.

The Cauchy-point condition (3.5) ensures that the approximate solution of the trust-region subproblem (2.2) satisfies the inequalities:

$$-\mathcal{Q}_j(s_j) \geq \tau\|\hat{g}_j\| \min\big(\delta_j, \|\hat{g}_j\|/\|\widehat{B}_j\|\big) \quad \text{and} \quad \|s_j\|_{T_j} \leq \delta_j. \tag{4.9}$$

(For a proof, see Powell [50]). This inequality plays a crucial role in the next lemma, which verifies that Algorithm 3.1 has property ($\mathbf{P}_1$).

**Lemma 4.3.** *Let $\{v_j\}$ be a sequence of iterates generated by* Algorithm 3.1 *such that $\{\widehat{B}_j\}$, $\{M_j\}$ and $\{M_j^{-1}\}$ are bounded. Assume that for each $j$, the step $s_j$ satisfies the Cauchy-point condition* (3.5) *and the step length $\alpha_j$ satisfies condition* (3.1)*. Then, if $\{\mathcal{M}(v_j)\}$ is bounded below and $\{\|\hat{g}_j\|\}$ is bounded away from zero, either $\delta_j \to 0$ and $\{v_j\}$ converges, or some $v_j$ satisfies the termination criteria and the algorithm terminates.*

*Proof.* A simple calculation gives

$$\sum_{l=0}^{j} \mathcal{M}(v_l) - \mathcal{M}(v_{l+1}) = \mathcal{M}(v_0) - \mathcal{M}(v_{j+1}),$$

which implies that if the sequence $\{\mathcal{M}(v_j)\}$ is bounded below, then the sequence of partial sums $\sum_{l=0}^{j} \mathcal{M}(v_l) - \mathcal{M}(v_{l+1})$ must be bounded above.

Let $\mathcal{S}$ denote the set of indices of the successful iterations, i.e.,

$$\mathcal{S} = \{\, j : \mathcal{M}(v_j) - \mathcal{M}(v_j + s_j) \geq -\eta_1 q_j(s_j) \,\}.$$

Since $\mathcal{M}(v_j) - \mathcal{M}(v_{j+1}) \geq 0$ for all $j$, the sum of a subset of the terms must be bounded, i.e., $\sum_{j\in\mathcal{S}} \mathcal{M}(v_j) - \mathcal{M}(v_{j+1}) < \infty$.

Every $s_j$ satisfies inequality (4.9), and so, for those iterates with $j \in \mathcal{S}$,

$$\mathcal{M}(v_j) - \mathcal{M}(v_j + s_j) \geq -\eta_1 q_j(s_j) \geq -\eta_1 \mathcal{Q}_j(s_j) \geq \eta_1 \tau \|\hat{g}_j\| \min\big(\delta_j, \|\hat{g}_j\|/\|\widehat{B}_j\|\big).$$

Summing over all successful iterations gives

$$\sum_{j\in\mathcal{S}} \mathcal{M}(v_j) - \mathcal{M}(v_{j+1}) \geq \tau\eta_1 \sum_{j\in\mathcal{S}} \|\hat{g}_j\| \min\big(\delta_j, \|\hat{g}_j\|/\|\widehat{B}_j\|\big), \tag{4.10}$$

and hence if $\|\hat{g}_j\|$ is bounded below and $\|\widehat{B}_j\|$ is bounded above, then $\sum_{j\in\mathcal{S}} \delta_j < \infty$.

Let $\bar{\alpha} = \nu\gamma_2 < 1$. Note that $\delta_{j+1} < \bar{\alpha}\delta_j$ whenever $j \notin \mathcal{S}$. If there are no successful iterations, then $\delta_j \leq \bar{\alpha}^{j-1}\delta_1$, in which case it is clear that $\sum_{j\to\infty} \delta_j < \infty$. Similarly, if $l$ is the last successful iteration, then for $j > l$, $\delta_j \leq \bar{\alpha}^{j-l-1}\delta_{l+1}$. Again, it follows that $\sum_{j\to\infty} \delta_j < \infty$.

Now suppose that there are an infinite number of successful iterations. Let $i$ be the index of any successful iteration such that the next iteration is unsuccessful. Let $k$ be an index such that $j \notin \mathcal{S}$ for $i+1 \leq j \leq k$. In other words $i+1, \ldots, k$ is the largest consecutive sequence of unsuccessful iterations starting at iteration $i+1$. For every index $j$ such that $i+1 \leq j \leq k$, it follows that $\delta_j \leq \bar{\alpha}^{j-i-1}\gamma_3\delta_i$ and hence

$$\sum_{j=i+1}^{k} \delta_j \leq \sum_{j=i+1}^{k} \bar{\alpha}^{j-i-1}\gamma_3\delta_i \leq \gamma_3\delta_i \sum_{j=1}^{\infty} \bar{\alpha}^j = \frac{\gamma_3}{1-\bar{\alpha}}\delta_i.$$

This leads to the inequality

$$\sum_{j\to\infty} \delta_j \leq \frac{\gamma_3}{1-\bar{\alpha}} \sum_{j\in\mathcal{S}} \delta_j < \infty,$$

and it follows immediately that $\sum_{j\to\infty}\delta_j < \infty$ and hence $\delta_j \to 0$.

The boundedness of the sequences $\{M_j\}$ and $\{M_j^{-1}\}$ and the implicit bound $\|x_{j+1} - x_j\|_{M_j} \leq \delta_j$ imposed by the trust-region constraint implies that $\{x_j\}$ converges to some limit $\bar{x}$. Since $\delta_j \to 0$, the updating rules for the trust-region radius will cause the choice $\alpha_j < 1$ to occur infinitely often. This, in turn, will cause the multipliers to be reset infinitely often using (3.4). The continuity of $\pi(x)$ implies that for any $\epsilon > 0$, there exists a sufficiently large positive integer $L$ such that for all $j > L$ for which $y_j = \pi_j$, it holds that $\|\pi(\bar{x}) - \pi_j\| < \epsilon/2$. The sequence $\|W_j^{-1/2}\|$ is bounded above and hence, without loss of generality, $L$ is also sufficiently large that $\sum_{j>L}\delta_j < \frac{1}{2}\inf_j \epsilon/\|W_j^{-1/2}\|$. But whenever $\alpha_j = 1$, $\|y_{j+1} - y_j\| < \|W_j^{-1/2}\|\delta_j$ and so we may conclude that for $j > L$, $\|y_j - \pi(\bar{x})\| < \epsilon$, and thus that $\{v_j\}$ converges. □

The next lemma establishes the conditions under which Algorithm 3.1 has property ($\mathbf{P}_2$).

**Lemma 4.4.** *Let $\{v_j\}$ be a sequence of iterates generated by* Algorithm 3.1 *such that $\{\widehat{B}_j\}$, $\{M_j\}$ and $\{M_j^{-1}\}$ are bounded. Assume that for each $j$, the step $s_j$ satisfies the Cauchy-point condition* (3.5) *and the step length $\alpha_j$ satisfies condition* (3.1)*. Then, if $\{\|\hat{g}_j\|\}$ is bounded away from zero and $\{v_j\}$ converges, then $\delta_j \not\to 0$.*

*Proof.* We use Taylor's Theorem to obtain the inequality

$$\begin{aligned}|\mathcal{M}(v_j + s_j) - \mathcal{M}(v_j) - q_j(s_j)| &\leq \|s_j\|_{T_j}\max_{0\leq\xi\leq 1}\|\hat{g}(v_j + \xi s_j) - \hat{g}(v_j)\| \\ &\quad + \tfrac{1}{2}\|s_j\|_{T_j}^2\|\widehat{B}_j\| \\ &\leq \delta_j\max_{0\leq\xi\leq 1}\|\hat{g}(v_j + \xi s_j) - \hat{g}(v_j)\| + \tfrac{1}{2}\delta_j^2\|\widehat{B}_j\|.\end{aligned}$$

Dividing both sides of this inequality by $|q_j(s_j)|$ and using the definition of $\rho_j$ gives

$$|\rho_j - 1| \leq \frac{\delta_j}{|q_j(s_j)|}\max_{0\leq\xi\leq 1}\|\hat{g}(v_j + \xi s_j) - \hat{g}(v_j)\| + \frac{\delta_j^2}{2|q_j(s_j)|}\|\widehat{B}_j\|. \tag{4.11}$$

If $\{\|\hat{g}_j\|\}$ is bounded below and $\{\widehat{B}_j\}$ is bounded, then (4.9) implies that for $\delta_j$ sufficiently small, the inequality $|q_j(s_j)| = -q_j(s_j) \geq \kappa\delta_j$ holds for some $\kappa > 0$. Because $\{v_j\}$ converges, all iterates lie in a compact region in which $\hat{g}(v)$ is continuous, and therefore uniformly continuous. We may conclude that if $\delta_j \to 0$, then

$$\max_{0\leq\xi\leq 1}\|\hat{g}(v_j + \xi s_j) - \hat{g}(v_j))\| \to 0.$$

Suppose that $\delta_j \to 0$. Inequality (4.11) then implies that $\rho_j \to 1$. However, the rules for modifying the trust-region radius imply that if $\rho_j \to 1$ then $\delta_j \not\to 0$. This is a contradiction and it must hold that $\delta_j \not\to 0$. □

The preceding lemmas yield the following theorem.

**Theorem 4.2.** *Let $\{v_j\}$ be a sequence of iterates generated by* Algorithm 3.1 *such that $\{\widehat{B}_j\}$, $\{M_j\}$ and $\{M_j^{-1}\}$ are bounded. Assume that for each $j$, the step $s_j$ satisfies the Cauchy-point condition* (3.5) *and the step length $\alpha_j$ satisfies condition* (3.1)*. If the sequence $\{\mathcal{M}(v_j)\}$ is bounded below, then either $\liminf_{j\to\infty} \|\hat{g}_j\| = 0$ or some $v_j$ satisfies the termination criterion and the algorithm terminates.* □

*4.2. Second-order analysis*

Further convergence results require the additional assumptions, possibly the weakest of which is that $\hat{g}(x, y)$ is uniformly continuous. Here we prefer to make the assumption that the sequence $\{x_j\}$ lies in a compact region, because $\|x_j\|$ can be monitored during the course of the computation.

**Lemma 4.5.** *Assume that the iterates $\{x_j\}$ generated by* Algorithm 3.1 *lie in a compact region. Then*

(a) *$\{c_i(x_j)\}$ and $\{y_{ij}\}$ are bounded above and bounded away from zero for $i \in \mathcal{I}$;*
(b) *$\{y_j\}$ is bounded;*
(c) *$\{(x_j, y_j)\}$ lies within a compact region; and*
(d) *$\{v_j + \alpha_j s_j\}$ lies within a compact region.*

*Proof.* Lemma 4.2 establishes that for $i \in \mathcal{I}$, $\{c_i(x_j)\}$ is bounded away from zero and that $\{y_{ij}\}$ is bounded above. Given the assumption that the iterates $\{x_j\}$ lie in a compact region, the sequence $\{c_i(x_j)\}$ is bounded above for all $i$, and in particular for $i \in \mathcal{I}$. Thus Lemma 4.2 also implies that $\{y_{ij}\}$ is bounded away from zero for $i \in \mathcal{I}$.

Similarly, since each $x_j$ lies in a compact set, the sequence $\{\mathcal{B}(x_j)\}$ of penalty-barrier function values (1.4) is bounded below by some $\bar{\mathcal{B}}$ (say). Omitting selected nonnegative terms from the merit function (1.8) and bounding $\{\mathcal{B}(x_j)\}$ from below by $\bar{\mathcal{B}}$ yields

$$\mathcal{M}(x_0, y_0) > \mathcal{M}(x_j, y_j) \geq \bar{\mathcal{B}} + \frac{1}{2\mu}\big(c_i(x_j) + \mu y_{ij}\big)^2 \quad \text{for} \quad i \in \mathcal{E}. \tag{4.12}$$

It follows that $\{y_{ij}\}$ is bounded for $i \in \mathcal{E}$. Norm equivalence in finite dimensional space implies that $\{y_j\}$ is also bounded.

Because the sequences $\{x_j\}$ and $\{y_j\}$ both lie within compact regions of finite dimensional space, the product sequence $\{(x_j, y_j)\}$ must lie within a compact region of the product space.

Let $s_j = (\Delta x_j, \Delta y_j)$ and note that $x_{j+1} = x_j + \alpha_j \Delta x_j$. Define $\zeta_{j+1} = y_j + \Delta y_j$. Then the bound (4.12) and the conclusions of this lemma apply to $(x_j, \zeta_j)$ as well as $(x_j, y_j)$. □

Note that the bound on $\{y_j\}$ is the bound on a sequence of iterates of Algorithm 3.1 for a fixed value of $\mu$. The Lagrange multipliers of (NP) need not be bounded.

**Corollary 4.2.** *Assume that the iterates* $\{x_j\}$ *generated by* Algorithm 3.1 *lie in a compact region, and that the sequences* $\{M_j\}$ *and* $\{M_j^{-1}\}$ *are bounded. Then*

(a) $\{W_j\}$, $\{W_j^{-1}\}$, $\{T_j\}$ *and* $\{T_j^{-1}\}$ *are bounded;*
(b) $\hat{g}_j = \Theta\big(\|g_j\|\big)$*; and*
(c) $\widehat{B}_j = \Theta\big(\|B_j\|\big)$ *with* $\lambda_{\min}(\widehat{B}_j) = \Theta\big(\lambda_{\min}(B_j)\big)$.

*Proof.* From the definitions (4.3) of $\hat{g}$ and $\widehat{B}$, we have

$$\hat{g}_j = T_j^{-\frac{1}{2}} g_j \quad \text{and} \quad \widehat{B}_j = T_j^{-\frac{1}{2}} B_j T_j^{-\frac{1}{2}}, \quad \text{with} \quad T_j = \operatorname{diag}(M_j, W_j).$$

The combined results of Corollary 4.1 and Lemma 4.5 imply that the diagonals of $\{W_j\}$ are bounded above and bounded away from zero. The result follows immediately. □

Corollary 4.2 implies that under the assumption that the iterates lie in a compact region, it is no longer necessary to distinguish between $g_j$ and $\hat{g}_j$ or $B_j$ and $\widehat{B}_j$ in the convergence results.

**Lemma 4.6.** *Assume that the iterates* $\{x_j\}$ *generated by* Algorithm 3.1 *lie in a compact region. Then the sequence* $\{(x_j, y_j)\}$ *lies in the interior of a region within which* $g(x, y)$ *is uniformly continuous. Moreover,* $B(x, y)$ *is also uniformly continuous in the same region and hence* $\{B_j\}$ *is bounded.*

*Proof.* As the sequences $\{c_i(x_j)\}$ and $\{y_{ij}\}$ are both bounded away from zero for $i \in \mathcal{I}$, all limit points of $\{(x_j, y_j)\}$ lie in a region within which $g(x, y)$ is continuous. Thus, because the closure of $\{(x_j, y_j)\}$ is compact, it follows that $\sup_j \|g(x_j, y_j)\| < \infty$. Let the value of this supremum be denoted by $L$. Let $\beta > 0$ be a constant for which $c_i(x_j) > \beta$ and $y_{ij} > \beta$ for $i \in \mathcal{I}$ and any $j$. Then for any $\epsilon > 0$, the set

$$\Omega = \{\, (x, y) : \|g(x, y)\| \leq L + \epsilon \text{ and } c_i(x) \geq \beta,\ y_i \geq \beta \,\}$$

is closed and the sequence $\{(x_j, y_j)\}$ lies entirely within its interior. Furthermore, the functions $g(x, y)$ and $B(x, y)$ are continuous at every point in $\Omega$. (The discontinuities occur at points for which $c_i(x) = 0$ or $y_i = 0$, but none of these points lie in $\Omega$.)

Because $\{(x_j, y_j)\}$ lies within a compact region in finite-dimensional space, there is a closed ball $\mathcal{D}$ centered at the origin that is large enough to contain $\{(x_j, y_j)\}$ entirely within its interior. It follows that the set $\Omega \cap \mathcal{D}$ is compact and $\{(x_j, y_j)\}$ lies in its interior. The function $g(x, y)$ is continuous, and hence uniformly continuous, in the set $\Omega \cap \mathcal{D}$. □

Under the assumption that $\{x_j\}$ lies in a compact region, we can prove strong convergence results involving the sequence $\{B_j\}$. In particular, it will be shown that if the trust-region subproblem is solved using the techniques described in Section 5, then there is a subsequence $\{(x_j, y_j)\}_{j \in \mathcal{S}}$ of $\{(x_j, y_j)\}$ for which $\lim_{j \in \mathcal{S}} \|g_j\| = 0$, and $\lim_{j \in \mathcal{S}} \lambda_{\min}(B_j) \geq 0$. It will also be shown that

this is a strong convergence result on $\{\nabla^2\mathcal{M}(x,y)\}$, for which $\{B_j\}$ is a sequence of approximations.

Algorithms for finding an approximate solution to the trust-region subproblem satisfying condition (3.6) are based on Theorem 2.1, which together with condition (3.6), provides the following useful bound.

**Corollary 4.3.** *If $s_j$ satisfies the trust-region termination condition* (3.6) *then* $-\mathcal{Q}(s_j) \geq \frac{1}{2}\tau\sigma_j\delta_j^2$. □

This result is used extensively in the convergence proof of Sorensen [54], which we follow broadly here. A major difference, however, is that the update to the trust-region radius is controlled by the ratio

$$\rho_j = (\mathcal{M}(v_j + s_j) - \mathcal{M}(v_j))/q_j(s_j)$$

instead of the ratio $(\mathcal{M}(v_j + s_j) - \mathcal{M}(v_j))/\mathcal{Q}_j(s_j)$ used in conventional trust-region algorithms. It will be shown that if $s_j$ satisfies (3.6) and $\delta_j \to 0$, then

$$|\rho_j - \mathcal{Q}_j(s_j)/q_j(s_j)| \to 0. \tag{4.13}$$

It must be emphasized that the relations derived in the proof of Lemma 4.4 are based on the counter-factual assumption that $\{\|\hat{g}_j\|\}$ is bounded away from zero, and that (4.13) holds when this assumption is not made. Thus, we require the following lemma to establish a lower bound on the ratio $\mathcal{Q}_j(s_j)/q_j(s_j)$.

**Lemma 4.7.** *Let $s_j$ be an approximate solution of the trust-region subproblem satisfying the termination condition* (3.6). *Then*

$$\frac{\mathcal{Q}_j(s_j)}{q_j(s_j)} \geq \frac{1}{2}\big(1 - \sqrt{1-\tau}\big). \tag{4.14}$$

*Proof.* If $s_j^T B_j s_j \leq 0$, the definition of $q_j(s_j)$ (3.2) implies that $\mathcal{Q}_j(s_j)/q_j(s_j) = 1$ and (4.14) follows from the trivial inequality $1 > \frac{1}{2}\big(1 - \sqrt{1-\tau}\big)$. For the remainder of the proof, it will be assumed that $s_j^T B_j s_j > 0$.

If $\mathcal{Q}_j^* = 0$, then $g_j = 0$, $B_j$ is positive semidefinite and we assume that the algorithm terminates. Otherwise, it must hold that $\mathcal{Q}_j^* < 0$, in which case the termination condition (3.6) for the trust-region subproblem guarantees that

$$0 > \tau\mathcal{Q}_j^* \geq \mathcal{Q}_j(s_j) = g_j^T s_j + \tfrac{1}{2}s_j^T B_j s_j \geq g_j^T s_j,$$

and $g_j^T s_j < 0$.

The required lower bound on $\mathcal{Q}_j(s_j)/q_j(\alpha s_j)$ will be derived in terms of the scalar $\beta$ that solves the problem

$$\underset{\alpha}{\text{minimize}} \;\; \mathcal{Q}_j(\alpha s_j) \quad \text{subject to} \;\; \|\alpha s_j\|_{T_j} \leq \zeta, \tag{4.15}$$

where $\zeta = \|s_j\|_{T_j} \leq \delta_j$. Under our assumption that $s_j^T B_j s_j > 0$, it must hold that $\beta = -g_j^T s_j / s_j^T B_j s_j$ if $-g_j^T s_j \leq s_j^T B_j s_j$, and $\beta = 1$ otherwise. In either case, $\beta$ must satisfy the inequalities $\beta \leq 1$ and $\beta \leq -g_j^T s_j / s_j^T B_j s_j$.

The termination condition (3.6) ensures that $\mathcal{Q}_j(s_j) \leq \tau\mathcal{Q}_j^* \leq \tau\mathcal{Q}_j(\beta s_j)$. Focusing on the inequality $\mathcal{Q}_j(s_j) \leq \tau\mathcal{Q}_j(\beta s_j)$ yields

$$\mathcal{Q}_j(s_j) \leq \tau\mathcal{Q}_j(\beta s_j) = \tau(\beta g_j^T s_j + \tfrac{1}{2}\beta^2 s_j^T B_j s_j) \leq \tfrac{1}{2}\tau\beta g_j^T s_j, \tag{4.16}$$

with the last inequality being a consequence of the bound $\beta \leq -g_j^T s_j / s_j^T B_j s_j$. Dividing each of these inequalities by $q_j(s_j) = g_j^T s_j < 0$, we arrive at the inequality

$$\mathcal{Q}_j(s_j)/q_j(s_j) \geq \tfrac{1}{2}\tau\beta, \tag{4.17}$$

and it remains to find a lower bound on $\beta$. The value of $\beta$ will depend upon whether $\beta$ is a constrained or unconstrained solution of (4.15). If $\beta = 1$, then $\mathcal{Q}_j(s_j)/q_j(s_j) \geq \frac{1}{2}\tau \geq \frac{1}{2}(1 - \sqrt{1-\tau})$, as required. On the other hand, if $\beta = -g_j^T s_j / s_j^T B_j s_j$, then the inequalities (4.16) imply that $\beta\mathcal{Q}_j(s_j) \leq \frac{1}{2}\tau\beta^2 g_j^T s_j$, which leads to the inequality

$$\beta g_j^T s_j + \tfrac{1}{2}\beta s_j^T B_j s_j \leq \tfrac{1}{2}\tau\beta^2 g_j^T s_j.$$

Substituting $-g_j^T s_j$ for $\beta s_j^T B_j s_j$ and canceling the (necessarily negative) term $g_j^T s_j$ gives $\frac{1}{2}\tau\beta^2 - \beta + \frac{1}{2} \leq 0$, which implies

$$\frac{1 - \sqrt{1-\tau}}{\tau} \leq \beta \leq \frac{1 + \sqrt{1-\tau}}{\tau}.$$

Using the lower bound on $\beta$ in (4.17) yields the result (4.14). □

**Theorem 4.3.** *Assume that the sequence* $\{x_j\}$ *generated by* Algorithm 3.1 *lies in a compact region, and that* $\lim_{j\to\infty} \|B_j - \nabla^2\mathcal{M}(v_j)\| = 0$. *Suppose that for each* $j$, *the step* $s_j$ *satisfies the termination criteria* (3.6) *with* $g_j^T s_j \leq 0$, *and the step length* $\alpha_j$ *satisfies the line search criterion* (3.1) *with* $\eta_1$ *chosen such that* $\eta_1 < \frac{1}{2}(1 - \sqrt{1-\tau})$. *Then either some* $v_j$ *satisfies the termination criteria and the algorithm terminates or* $\limsup_{j\to\infty} \lambda_{\min}\big(\nabla^2\mathcal{M}(v_j)\big) \geq 0$.

*Proof.* Let $\mathcal{S} = \{\, j : \big(\mathcal{M}(v_j + s_j) - \mathcal{M}(v_j)\big)/q_j(s_j) \geq \eta_1 \,\}$ denote the set of indices of the successful iterations. The definition (3.2) of $q_j(s_j)$ implies that $q_j(s_j) \leq \mathcal{Q}_j(s_j)$, which allows us to extend the inequality of Corollary 4.3 so that

$$-q_j(s_j) \geq \frac{\tau}{2}\sigma_j\delta_j^2. \tag{4.18}$$

It follows that for the successful iterations

$$\mathcal{M}(v_j) - \mathcal{M}(v_j + s_j) \geq -\eta_1 q_j(s_j) \geq \frac{\tau}{2}\eta_1\sigma_j\delta_j^2. \tag{4.19}$$

Suppose that $\{\sigma_j\}$ is bounded away from zero. If the set $\mathcal{S}$ is infinite, then because $\{\mathcal{M}(v_j)\}$ is decreasing and bounded below, the inequality (4.19) implies $\lim_{j\in\mathcal{S},\ j\to\infty} \delta_j = 0$. If $m \notin \mathcal{S}$, let $l < m$ be the largest predecessor of $m$ that is in $\mathcal{S}$. In other words, $l \in \mathcal{S}$ and $j \notin S$ for $l < j < m$. Then $\delta_m \leq \gamma_3\delta_l$. It follows that if $\mathcal{S}$ is infinite, then $\lim_{j\to\infty} \delta_j = 0$. On the other hand, if $\mathcal{S}$ is finite, then

for sufficiently large $j$, $\delta_{j+1} \leq \gamma_2 \nu \delta_j$. Because $\gamma_2 \nu < 1$, for this case we also find that $\lim_{j\to\infty} \delta_j = 0$.

We now show that if $\lim_{j\to\infty} \delta_j = 0$ and $\eta_1$ is chosen such that $\eta_1 < \frac{1}{2}(1 - \sqrt{1-\tau})$, then $\rho_j = (\mathcal{M}(v_j + s_j) - \mathcal{M}(v_j))/q_j(s_j) \geq \eta_1$ for $j$ sufficiently large. By Taylor's theorem,

$$\begin{aligned}
&|\mathcal{M}(v_j + s_j) - \mathcal{M}(v_j) - \mathcal{Q}_j(s_j)| \\
&= |\mathcal{M}(v_j + s_j) - \mathcal{M}(v_j) - g_j^T s_j - \tfrac{1}{2} s_j^T B_j s_j| \\
&\leq \tfrac{1}{2}\|s_j\|^2 \big( \max_{0\leq\xi\leq 1} \|\nabla^2\mathcal{M}(v_j + \xi s_j) - \nabla^2\mathcal{M}(v_j)\| + \|B_j - \nabla^2\mathcal{M}(v_j)\| \big) \\
&\leq \tfrac{1}{2}\|T_j^{-1}\|^2 \delta_j^2 \big( \max_{0\leq\xi\leq 1} \|\nabla^2\mathcal{M}(v_j + \xi s_j) - \nabla^2\mathcal{M}(v_j)\| + \|B_j - \nabla^2\mathcal{M}(v_j)\| \big).
\end{aligned}$$

Dividing both sides of this expression by $|q_j(s_j)|$ and using the inequality (4.18) yields

$$\begin{aligned}
&|\rho_j - \mathcal{Q}_j(s_j)/q_j(s_j)| \\
&\qquad \leq \frac{\|T_j^{-1}\|^2}{\tau\sigma_j} \big( \max_{0\leq\xi\leq 1} \|\nabla^2\mathcal{M}(v_j + \xi s_j) - \nabla^2\mathcal{M}(v_j)\| + \|B_j - \nabla^2\mathcal{M}(v_j)\| \big).
\end{aligned}$$

By assumption, the sequence $\{\sigma_j\}$ is bounded away from zero, $\{T_j^{-1}\}$ is bounded and $\lim_{j\to\infty} \|B_j - \nabla^2\mathcal{M}(v_j)\| = 0$. It follows that if $\delta_j \to 0$, then $\|s_j\| \to 0$ and by the uniform continuity of $\nabla^2\mathcal{M}(v)$ it must hold that

$$\lim_{j\to\infty} |\rho_j - \mathcal{Q}_j(s_j)/q_j(s_j)| = 0. \tag{4.20}$$

This result and Lemma 4.7 imply that if $\eta_1 < \frac{1}{2}(1 - \sqrt{1-\tau})$, then $\rho_j \geq \eta_1$ for all $j$ sufficiently large. However, the updating rules of Algorithm 3.1 choose $\delta_{j+1} \geq \delta_j$ if $\rho_j \geq \eta_1$, and so it is not possible for $\delta_j \to 0$. This contradiction implies that our assumption that $\sigma_j$ is bounded away from zero is false, and since $\sigma_j \geq -\lambda_{\min}(B_j)$, it must hold that $\limsup_{j\to\infty} \lambda_{\min}(B_j) \geq 0$. The assumption that $\|\nabla^2\mathcal{M}(v_j) - B_j\| \to 0$ now implies that $\limsup_{j\to\infty} \lambda_{\min}\big(\nabla^2\mathcal{M}(v_j)\big) \geq 0$. □

It remains to be shown that $\lim_{j\to\infty} \|B_j - \nabla^2\mathcal{M}(v_j)\| = 0$. The following lemma will be of use.

**Lemma 4.8.** *Assume that the sequence* $\{x_j\}$ *generated by* Algorithm 3.1 *lies in a compact region. If* $\lim_{j\to\infty} \|g_j\| = 0$ *then* $\lim_{j\to\infty} \|B_j - \nabla^2\mathcal{M}(v_j)\| = 0$.

*Proof.* Corollary 4.2 implies that the sequences $\{W_j\}$ and $\{W_j^{-1}\}$ are bounded. It follows from the definition of $g_j$ that if $\lim_{j\to\infty} \|g_j\| = 0$, then $y_j \to \pi_j$. However, Lemma 4.6 shows that the iterates lie in a region in which $B(x, y)$ is uniformly continuous, and so the identity $B(x, \pi) = \nabla^2\mathcal{M}(x, \pi)$ gives the required limit $\lim_{j\to\infty} \|B_j - \nabla^2\mathcal{M}(v_j)\| = 0$. □

*4.3. Convergence of the gradient*

The proof that $\lim_{j\to\infty} \|g_j\| = 0$ is arguably the most subtle part of the analysis. It is an extension of Thomas' [55] original proof for the conventional trust-region method. The use of a line search complicates the proof considerably.

In this section, we modify the notation slightly in order to simplify the exposition. The successful iterations of Algorithm 3.1 do not involve a line search, and as a consequence, a step length is defined only for those iterations for which the condition $\mathcal{M}(v_j + s_j) - \mathcal{M}(v_j))/q_j(s_j) < \eta_1$ holds. For the purposes of this section, we will associate a step length $\alpha_j$ with *every* iteration, with the assumption that $\alpha_j = 1$ for the successful iterations.

The main result requires five lemmas.

**Lemma 4.9.** *Assume that the sequence $\{x_j\}$ generated by Algorithm* 3.1 *lies in a compact region. Then the sequence $\{\delta_j\}$ is bounded above.*

*Proof.* From Lemma 4.5, both the sequences $\{v_j\}$ and $\{v_j + \alpha_j s_j\}$ lie in a compact region. Thus, there is some $\Delta > 0$ such that $\|\alpha_j s_j\| \leq \Delta$ for all $j$. But then, since the rule for updating the trust-region radius is to choose $\delta_{j+1} \leq \max\left(\delta_j, \gamma_3\|\alpha_j s_j\|_{T_j}\right)$, we find that $\delta_j \leq \max\left(\delta_0, \gamma_3\Delta \sup_i\{\|T_i\|\}\right)$ for all $j$. By Corollary 4.2 the sequence $\{T_j\}$ is bounded, and so the sequence $\{\delta_j\}$ is bounded above. □

**Lemma 4.10.** *Let $\{v_j\}$ be the sequence of iterates generated by Algorithm* 3.1 *and let $\{\alpha_j\}$ be the corresponding sequence of step lengths. If $\bar{\alpha}$ is any number in the interval $(0, 1/\nu)$, then for any iteration for which $\alpha_j \geq \bar{\alpha}$, it holds that*

$$\mathcal{M}(v_j) - \mathcal{M}(v_j + \alpha_j s_j) \geq \eta_1\tau\bar{\alpha}\|\hat{g}_j\| \min\left(\alpha_j\delta_j, \alpha_j\|\hat{g}_j\|/\|\widehat{B}_j\|\right).$$

*Proof.* A combination of the inequality $-q_j(s_j) \geq -\mathcal{Q}_j(s_j)$, the Cauchy-point condition (3.5) and the Powell inequality (4.9) yield

$$-q_j(s_j) \geq \tau\|\hat{g}_j\| \min\left(\delta_j, \|\hat{g}_j\|/\|\widehat{B}_j\|\right).$$

Consider the set $\mathcal{A} = \{\, j : \alpha_j \geq \bar{\alpha}\,\}$. First, the inequality $\bar{\alpha} < 1/\nu \leq 1$ may be used to obtain an expression for $q_j(\alpha_j s_j)$ in terms of $q_j(s_j)$ for all $j \in \mathcal{A}$. If $s_j^T B_j s_j \geq 0$, then $-q_j(\alpha_j s_j) = -\alpha_j g_j^T s_j \geq -\bar{\alpha}\alpha_j q_j(s_j)$. Alternatively, if $s_j^T B_j s_j < 0$, then

$$-q_j(\alpha_j s_j) = -\alpha_j g_j^T s_j - \tfrac{1}{2}\alpha_j^2 s_j^T B_j s_j \geq -\bar{\alpha}\alpha_j g_j^T s_j - \tfrac{1}{2}\bar{\alpha}\alpha_j s_j^T B_j s_j = -\bar{\alpha}\alpha_j q_j(s_j).$$

In either case, $\alpha_j s_j$ satisfies the Armijo-style condition (3.1) and

$$\mathcal{M}(v_j) - \mathcal{M}(v_j + \alpha_j s_j) \geq \eta_1\tau\bar{\alpha}\|\hat{g}_j\| \min\left(\alpha_j\delta_j, \alpha_j\|\hat{g}_j\|/\|\widehat{B}_j\|\right)$$

for all $j \in \mathcal{A}$. □

The next lemma considers the properties of a consecutive sequence of iterates in which $\|g(v_j)\|$ decreases by a positive amount at least as large as a positive quantity $\epsilon_1 - \epsilon_2$.

**Lemma 4.11.** *Assume that the sequence* $\{x_j\}$ *generated by Algorithm* 3.1 *lies in a compact region. Choose* $\epsilon_1 > \epsilon_2 > 0$. *For every* $\epsilon_3 > 0$ *there is an* $L$ *sufficiently large that* $\sum_{j=p}^{q-1} \|\alpha_j s_j\| < \epsilon_3$ *for all indices* $q > p \geq L$ *for which* $\|g(v_p)\| > \epsilon_1$, $\|g(v_j)\| > \epsilon_2$ *for consecutive indices* $j = p,\, p+1,\, \ldots,\, q-1$, *and* $\|g(v_q)\| < \epsilon_2$.

*Proof.* Note that if $\|g(v_j)\| > \epsilon_1$ holds only finitely often, then the lemma is trivially true. Furthermore, as $\liminf_{j\to\infty} \|g(v_j)\| = 0$, it must hold that for every iteration $p$ such that $\|g(v_p)\| > \epsilon_1$, there must exist a subsequent iteration $q$ such that $\|g(v_q)\| < \epsilon_2$.

Let $\bar{\alpha}$ be any number in the interval $(0, 1/\nu)$. Lemma 4.10 implies that for every index in the set $\mathcal{A} = \{\, j : \alpha_j \geq \bar{\alpha} \,\}$, we have

$$\mathcal{M}(v_j) - \mathcal{M}(v_j + \alpha_j s_j) \geq \eta_1 \tau \bar{\alpha}^2 \|\hat{g}(v_j)\| \min\big(\delta_j, \|\hat{g}(v_j)\|/\|\widehat{B}_j\|\big).$$

Let $\mathcal{G}(p) = \{\, j : j \geq p \text{ and } \|g(v_j)\| > \epsilon_2 \,\}$. Because $\sum_{j=1}^{\infty} \mathcal{M}(v_j) - \mathcal{M}(v_{j+1}) < \infty$,

$$\sum_{j\in\mathcal{A}\cap\mathcal{G}(p)} \|\hat{g}(v_j)\| \min\big(\delta_j, \|\hat{g}(v_j)\|/\|\widehat{B}_j\|\big) < \infty.$$

By Corollary 4.2, $\widehat{B}_j = \Theta\big(\|B_j\|\big)$ and $\hat{g}_j = \Theta\big(\|g_j\|\big)$. As $\{B_j\}$ is bounded, and $\|g(v_j)\| > \epsilon_2$ for $j \in \mathcal{G}(p)$, it follows that

$$\sum_{j\in\mathcal{A}\cap\mathcal{G}(p)} \delta_j < \infty. \tag{4.21}$$

Let $\mathcal{J}$ denote the sequence of iteration indices $\{p,\, p+1,\, \ldots,\, q-1\}$. Let $\{j_k\}_{k=1}^{r}$ denote the subsequence of $\mathcal{J}$ with indices in $\mathcal{A}$. We now partition $\mathcal{J}$ into $r+1$ nonoverlapping subsequences $\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_r$ with $\mathcal{P}_0 = \{p, p+1, \ldots, j_1 - 1\}$, $\mathcal{P}_k = \{j_k, j_k+1, \ldots, j_{k+1}-1\}$, $k = 1, 2, \ldots, r-1$ and $\mathcal{P}_r = \{j_r, j_r+1, \ldots, q-1\}$. Note that if the first index $p$ is in $\mathcal{A}$, then $\mathcal{P}_0$ is empty. Otherwise none of the indices of $\mathcal{P}_0$ will be in $\mathcal{A}$. For $k > 0$, the sequence $\mathcal{P}_k$ starts with an index in $\mathcal{A}$, followed by a (possibly empty) sequence of indices that are not in $\mathcal{A}$. These definitions allow us to write the quantity to be bounded as

$$\sum_{j=p}^{q-1} \|\alpha_j s_j\| = \sum_{j\in\mathcal{P}_0} \|\alpha_j s_j\| + \sum_{k=1}^{r} \sum_{j\in\mathcal{P}_k} \|\alpha_j s_j\|. \tag{4.22}$$

First we estimate the quantity $\sum_{j\in\mathcal{P}_0} \|\alpha_j s_j\|$. If the set $\mathcal{P}_0$ is empty, then $\sum_{j\in\mathcal{P}_0} \|\alpha_j s_j\| = 0$. Otherwise, $\alpha_j < \bar{\alpha}$ for every $j \in \mathcal{P}_0$ and the rules for updating the trust-region radius give $\delta_{j+1} \leq \nu\bar{\alpha}\delta_j$. This gives the sequence of inequalities

$$\sum_{j\in\mathcal{P}_0} \|\alpha_j s_j\| \leq \sum_{j\in\mathcal{P}_0} \bar{\alpha}\delta_j \leq \bar{\alpha}\delta_p \sum_{i=0}^{\infty} (\nu\bar{\alpha})^i = \bar{\alpha}\Big(\frac{1}{1-\nu\bar{\alpha}}\Big)\delta_p \leq \bar{\alpha}\Big(\frac{1}{1-\nu\bar{\alpha}}\Big)\delta_{\max},$$

where $\delta_{\max}$ is the bound on $\{\delta_j\}$ established in Lemma 4.9. The scalar $\bar{\alpha}$ is arbitrary and may be chosen sufficiently small that

$$\bar{\alpha}\Big(\frac{1}{1-\nu\bar{\alpha}}\Big)\delta_{\max} \leq \tfrac{1}{2}\epsilon_3, \tag{4.23}$$

for any $\epsilon_3 > 0$.

To estimate the terms in (4.22) involving the indices in $\mathcal{P}_k$ for $k > 0$, we derive the following bound on the sum of trust-region radii:

$$\sum_{j \in \mathcal{P}_k} \delta_j \leq \Big(1 + \frac{\gamma_3}{1 - \nu\bar{\alpha}}\Big)\delta_{j_k}. \tag{4.24}$$

For $j \in P_k \setminus \{j_k\}$, i.e., for every element of $P_k$ except the first, $\alpha_j < \bar{\alpha}$ and the rules for updating the trust-region radius give $\delta_{j+1} \leq \nu\bar{\alpha}\delta_j$. Thus,

$$\sum_{j \in P_k \setminus \{j_k\}} \delta_j \leq \delta_{j_k+1} \sum_{i=0}^{\infty} (\nu\bar{\alpha})^i = \Big(\frac{1}{1 - \nu\bar{\alpha}}\Big)\delta_{j_k+1}. \tag{4.25}$$

At iteration $j_k$, it is possible that the trust-region will increase, so the appropriate bound on $\delta_{j_k+1}$ is $\delta_{j_k+1} \leq \gamma_3\delta_{j_k}$. Combining this bound with (4.25) yields (4.24).

From the definition of $\mathcal{P}_k$ we have

$$\sum_{j \in \mathcal{P}_k} \|\alpha_j s_j\| \leq \sum_{j \in \mathcal{P}_k} \delta_j \leq \Big(1 + \frac{\gamma_3}{1 - \nu\bar{\alpha}}\Big)\delta_{j_k}.$$

By construction, $j_k \in \mathcal{A} \cap \mathcal{G}(p)$ and we may sum the contributions from each of the subsequences to give

$$\sum_{k=1}^{r} \sum_{j \in \mathcal{P}_k} \|\alpha_j s_j\| \leq \Big(1 + \frac{\gamma_3}{1 - \nu\bar{\alpha}}\Big) \sum_{k=1}^{r} \delta_{j_k} \leq \Big(1 + \frac{\gamma_3}{1 - \nu\bar{\alpha}}\Big) \sum_{j \in \mathcal{A} \cap \mathcal{G}(p)} \delta_j.$$

Given the value of $\bar{\alpha}$ determined by (4.23), the bound (4.21) implies that we may choose $p$ sufficiently large that

$$\Big(1 + \frac{\gamma_3}{1 - \nu\bar{\alpha}}\Big) \sum_{j \in \mathcal{A} \cap \mathcal{G}(p)} \delta_j < \tfrac{1}{2}\epsilon_3. \tag{4.26}$$

But then, from (4.23) and (4.26), we have

$$\sum_{j=p}^{q-1} \|\alpha_j s_j\| = \sum_{j \in \mathcal{P}_0} \|\alpha_j s_j\| + \sum_{k=1}^{r} \sum_{j \in \mathcal{P}_k} \|\alpha_j s_j\| \leq \tfrac{1}{2}\epsilon_3 + \tfrac{1}{2}\epsilon_3 = \epsilon_3,$$

where $\epsilon_3$ may be chosen to be arbitrarily small. □

The next lemma concerns the limiting behavior of the gradient of the barrier-penalty function, i.e., $\nabla\mathcal{B}(x_j) = \nabla f(x_j) - J(x_j)^T\pi(x_j)$, where $\pi(x_j)$ is the vector of primal multipliers (see (1.4) and (1.5)).

**Lemma 4.12.** *Assume that the sequence $\{x_j\}$ generated by Algorithm 3.1 lies in a compact region. Then $\lim_{j\to\infty} \|\nabla\mathcal{B}(x_j)\| = 0$.*

*Proof.* First it must be shown that $\liminf_{j\to\infty} \|\nabla\mathcal{B}(x_j)\| = 0$. Theorem 4.2 and Corollary 4.2 establish that $\liminf_{j\to\infty} \|g(v_j)\| = 0$. It follows from the definition of $g(v_j)$ that there must be a subsequence for which both $y_j \to \pi_j$ and $\nabla\mathcal{B}(x_j) + J_j^T(y_j - \pi_j) \to 0$. The assumption that $\{x_j\}$ lies in a compact region implies that $\{J_j\}$ is bounded and so $\liminf_{j\to\infty} \|\nabla\mathcal{B}(x_j)\| = 0$ as required.

If $\|\nabla\mathcal{B}(x_j)\|$ does not converge to zero, there must exist an $\epsilon > 0$, such that $\|\nabla\mathcal{B}(x_j)\| > \epsilon$ infinitely often. Let $p$ be an index such that $\|\nabla\mathcal{B}(x_p)\| > \epsilon$ and let $q$ be the smallest index larger than $p$ for which $\|\nabla\mathcal{B}(x_p)\| < \frac{1}{2}\epsilon$. The definition of $g(v_j)$ is such that $\|g(v_j)\| \geq \|\nabla\mathcal{B}(x_j)\|$, and Lemma 4.11 can be applied to conclude that $p$ may be chosen sufficiently large to make $\sum_{j=p}^{q-1} \|\alpha_j s_j\|$ (and hence $\|x_p - x_q\|$) arbitrarily small. The uniform continuity of $\nabla\mathcal{B}(x)$ implies that $p$ can be chosen sufficiently large to make $\|\nabla\mathcal{B}(x_q) - \nabla\mathcal{B}(x_p)\| < \frac{1}{2}\epsilon$. This is a contradiction because

$$\tfrac{1}{2}\epsilon < \|\nabla\mathcal{B}(x_q)\| - \|\nabla\mathcal{B}(x_p)\| \leq \|\nabla\mathcal{B}(x_q) - \nabla\mathcal{B}(x_p)\|.$$

We conclude that $\lim_{j\to\infty} \|\nabla\mathcal{B}(x_j)\| = 0$. □

The merit function (1.8) may be written as $\mathcal{M}(x, y) = \mathcal{B}(x) + \Psi(x, y)$, where $\mathcal{B}(x)$ is the conventional barrier-penalty function (1.4) and $\Psi(x, y)$ is the proximity term

$$\Psi(x, y) = -\mu \sum_{i\in\mathcal{I}} \Big( \ln\Big(\frac{c_i(x)y_i}{\mu}\Big) + \frac{\mu - c_i(x)y_i}{\mu}\Big) + \frac{1}{2\mu} \sum_{i\in\mathcal{E}} \big(c_i(x) + \mu y_i\big)^2.$$

The next lemma concerns the behavior of this proximity term when the norm of the merit function gradient behaves nonmonotonically in the limit. In particular, it is shown that $\Psi(x, y)$ must increase by at least a fixed value if the merit gradient norm increases from a small value.

**Lemma 4.13.** *Assume that* $\liminf_{j\to\infty} \|g(v_j)\| = 0$ *and that there exists a positive number* $\epsilon_1$ *such that the relation* $\|g(v_j)\| > \epsilon_1$ *holds infinitely often. Then there exists an* $\omega > 0$*, a positive* $\epsilon_2$ *sufficiently small and an index* $L$ *sufficiently large that if* $q > p \geq L$ *with* $\|g(v_q)\| > \epsilon_1 > \epsilon_2 > \|g(v_p)\|$*, then* $\Psi(x_q, y_q) \geq \Psi(x_p, y_p) + \omega$.

*Proof.* Theorem 4.1 and Corollary 4.2 imply that $g_{n+i}(v_j) = \Theta\big(|r_i(v_j)|\big)$, where

$$r_i(v_j) = \begin{cases} c_i(x_j) + \mu y_{ij} & \text{for } i \in \mathcal{E}, \\ c_i(x_j)y_{ij} - \mu & \text{for } i \in \mathcal{I}. \end{cases}$$

It follows from the assumption $\|g(v_p)\| < \epsilon_2$ and the definition of $\Psi(x, y)$ in terms of the elements of $r(v_j)$ that we may choose $\epsilon_2$ to make $\Psi(x, y)$ arbitrarily close to zero.

From the definition of $g(v)$ (1.9) we have

$$g(v_j) = \begin{pmatrix} \nabla f_j - J_j^T(2\pi_j - y_j) \\ W_j(y_j - \pi_j) \end{pmatrix} = \begin{pmatrix} \nabla\mathcal{B}(x_j) + J_j^T(y_j - \pi_j) \\ W_j(y_j - \pi_j) \end{pmatrix},$$

and hence taking norms and using the triangle inequality gives

$$\|A_j(y_j - \pi_j)\| \geq |\, \|g(v_j)\| - \|\nabla \mathcal{B}(x_j)\| \,|, \quad \text{where} \quad A_j = \begin{pmatrix} J_j^T \\ W_j \end{pmatrix}.$$

Lemma 4.12 indicates that we may choose $L$ sufficiently large that $\|\nabla \mathcal{B}(x_j)\| < \epsilon_2$, and thus $\|A_j(y_j - \pi_j)\| \geq \epsilon_1 - \epsilon_2$. Since $\|a\| \leq \sqrt{n+m}\,\|a\|_\infty$ for any $(n+m)$-vector $a$, we may infer that at least one element of $A_j(y_j - \pi_j)$ must have magnitude at least $(\epsilon_1 - \epsilon_2)/\sqrt{n+m}$. This implies that the infinity norm of $W_j(y_j - \pi_j)$ must satisfy

$$\|W_j(y_j - \pi_j)\|_\infty \geq \gamma_j \frac{\epsilon_1 - \epsilon_2}{\sqrt{n+m}},$$

where $\gamma_j = \min\{1, \lambda_{\min}(W_j)/(\sqrt{m}\|J_j\|)$. But $\{\gamma_j\}$ is bounded away from zero. This implies that $\Psi(x_q, y_q)$ must have some minimum value. By choosing $\epsilon_2$ sufficiently small, we may make this minimum value greater than the maximum value for $\Psi(x_p, y_p)$. □

Now we come to the main result of this section.

**Theorem 4.4.** *Assume that the sequence $\{x_j\}$ generated by Algorithm 3.1 lies in a compact region. Then either some $v_j$ satisfies the termination criteria and the algorithm terminates or $\lim_{j\to\infty} \|g(v_j)\| = 0$.*

*Proof.* Suppose $\|g(v_j)\|$ does not converge to zero. Let $\epsilon_1 > 0$ be a number for which $\|g(v_j)\| > \epsilon_1$ infinitely often.

Given any $\epsilon_2$ such that $\epsilon_1 > \epsilon_2 > 0$, let $p$ be any index such that $\|g(v_p)\| < \epsilon_2$, and let $q$ denote the next index greater than $p$ for which $\|g(v_q)\| > \epsilon_1$. Similarly, let $r$ be the next index greater than $q$ such that $\|g(v_r)\| < \epsilon_2$. We may apply Lemma 4.13 to assert that for $p$ sufficiently large, the proximity term satisfies $\Psi(x_q, y_q) \geq \Psi(x_p, y_p) + \omega$. The property that the merit function decreases monotonically implies that $\mathcal{M}(x_q, y_q) < \mathcal{M}(x_p, y_p)$ and it must hold that $\mathcal{B}(x_q) < \mathcal{B}(x_p) - \omega$. But then by Lemma 4.11, we may choose $p$ large enough to make $\sum_{j=q}^{r} \|\alpha_j s_j\|$ arbitrarily small. Thus, since $\mathcal{B}(x)$ is uniformly continuous, it must hold that for all sufficiently large choices of $p$, $|\mathcal{B}(x_r) - \mathcal{B}(x_q)| < \frac{1}{2}\omega$ and hence

$$\mathcal{B}(x_r) < \mathcal{B}(x_p) - \tfrac{1}{2}\omega.$$

It follows that each time $\|g(v_j)\|$ increases from a value less than $\epsilon_2$ to a value greater than $\epsilon_1$ and then decreases again to a value less than $\epsilon_2$, the barrier-penalty function must decrease by at least a constant factor. As $\mathcal{B}(x)$ is bounded below in a compact region, the value of $\|g(v_j)\|$ can exceed $\epsilon_1$ only finitely often. The assumption that $\epsilon_1$ is arbitrary implies that $\lim_{j\to\infty} \|g(v_j)\| = 0$, as required. □

## 5. Solving the trust-region subproblem

The algorithm for the trust-region subproblem (2.2) is based on the method of Moré and Sorensen [45], which is in turn a modification of an algorithm by Gay [30]. In this section, we briefly describe our algorithm and discuss how it differs from the Moré-Sorensen approach. In the subsections that follow, we omit the subscript $j$ when considering the details associated with a single trust-region subproblem.

### *5.1. Background*

The necessary and sufficient conditions of Theorem 2.1 state that $s$ solves the trust-region subproblem if and only if $\|s\|_T \leq \delta$ and there exists a unique non-negative scalar $\sigma^*$ such that

$$(B + \sigma^* T)s = -g \quad \text{and} \quad \sigma^*\big(\delta - \|s\|_T\big) = 0. \tag{5.1}$$

Let $\psi(\sigma)$ denote the univariate function

$$\psi(\sigma) = \frac{1}{\delta}\big(\|p_\sigma\|_T - \delta\big), \quad \text{where } p_\sigma \text{ satisfies} \quad (B + \sigma T)p_\sigma = -g. \tag{5.2}$$

It follows that if $\lambda$ is the smallest eigenvalue of $\widehat{B}$, i.e., $\lambda = \lambda_{\min}(T^{-1/2}BT^{-1/2})$, then for any $\sigma > -\lambda$ the matrix $B + \sigma T$ is positive definite and $\psi(\sigma)$ is well-defined. Clearly, if $\sigma$ is a positive zero of $\psi(\sigma)$, then $s = p_\sigma$ is a global solution to the trust-region subproblem. This result is the basis of the method of Hebden [40] for the case $T = I$, which employs Newton's method to find an approximate zero of $\|p_\sigma\|_T - \delta$. In order to avoid difficulties associated with the singularities of $\|p_\sigma\|_T - \delta$, Reinsch [52] suggests an alternative approach based on finding a zero of

$$\phi(\sigma) = \frac{1}{\delta} - \frac{1}{\|p_\sigma\|_T}. \tag{5.3}$$

This function is also used in the safeguarded Newton method of Moré and Sorensen, which is the basis of the method considered here.

The main properties of $\phi(\sigma)$ are stated without proof in the following lemma. (For details, see, e.g., Gay [30], and Moré and Sorensen [45].)

**Lemma 5.1.** *Assume that the vector $g$ is nonzero. The function $\phi(\sigma)$ has the following properties:*

(a) *$\phi(\sigma)$ is twice continuously differentiable on $(-\lambda, \infty)$;*
(b) *$\phi(\sigma)$ is strictly decreasing and strictly convex on $(-\lambda, \infty)$;*
(c) *if $\lim_{\sigma \to -\lambda_+} \phi(\sigma) > 0$, then $\phi(\sigma)$ has a unique zero in $(-\lambda, \infty)$;*
(d) *$\lim_{\sigma \to -\lambda_+} \phi(\sigma) \leq 0$ if and only if the linear system $(B - \lambda T)p = -g$ is compatible.* □

It is straightforward to verify that if $(B - \lambda T)p = -g$ is compatible then the vector

$$\bar{p} = \lim_{\sigma \to -\lambda} -(B + \sigma T)^{-1} g, \tag{5.4}$$

exists and has finite norm. If $\|\bar{p}\|_T < \delta$, then $\phi(\sigma)$ has no zero in $(-\lambda, \infty)$ and there are two possibilities. If $\lambda$ is positive then the pair $(\sigma^*, s)$ such that $\sigma^* = 0$ and $s = -B^{-1}g$ satisfies the optimality conditions (5.1) because $\|s\|_T < \|\bar{p}\|_T < \delta$. Otherwise, if $\lambda$ is negative or zero, then there is a null vector $z$ for $B - \lambda T$ such that

$$(B - \lambda T)(\bar{p} + z) = -g \quad \text{and} \quad \|\bar{p} + z\|_T = \delta.$$

In this case, the optimality conditions (5.1) are satisfied by the pair $(\sigma^*, s)$ such that $\sigma^* = -\lambda$ and $s = \bar{p} + z$. Moré and Sorensen call the case in which $\sigma^* = -\lambda$ the "hard case".

### *5.2. Approximate solutions of the trust-region subproblem*

The Moré-Sorensen method uses a safeguarded Newton iteration to find an approximate zero of $\phi(\sigma)$. This generates a nonnegative sequence $\{\sigma_i\}$ and an associated sequence of vectors $\{p_i\}$ such that $B + \sigma_i T$ is positive semidefinite and $(B + \sigma_i T)p_i = -g$. For a given $\sigma$, the functions $\phi(\sigma)$ and $\psi(\sigma)$ have the same sign. Moreover, $|\psi(\sigma)|$ is a measure of the accuracy of $\sigma$ as an approximate zero of $\phi$. Let $\varepsilon$ be a fixed scalar such that $0 < \varepsilon < 1$. Given the sequence $\{\sigma_i\}$, the scalar $\sigma = \sigma_i$ is considered an approximate zero of $\phi(\sigma)$ (with $s = p_i$ the associated step) if the condition

$(\mathbf{C}_1)$ $|\psi(\sigma_i)| \leq \varepsilon$.

is satisfied. In addition to this convergence condition, there are two situations in which the Newton iteration is terminated prior to convergence:

$(\mathbf{T}_1)$ if $\sigma_i = 0$ and $\psi(\sigma_i) < \varepsilon$; or
$(\mathbf{T}_2)$ if $\sigma_i > 0$, $\psi(\sigma_i) < -\varepsilon$ and there exists a sufficiently accurate approximate null-vector $z_i$ of $B + \sigma_i T$ (see the condition (5.6) below).

When termination occurs because of condition $(\mathbf{T}_1)$, the scalar $\sigma = 0$ and vector $s = p_i$ satisfy the optimality conditions of Theorem 2.1. For condition $(\mathbf{T}_2)$, the vector $p_i$ associated with the final $\sigma_i$ is an approximation to $\bar{p}$ (see (5.4)) and the approximate null vector $z_i$ is scaled so that $\|p_i + z_i\|_T = \delta$. In this case, we define $\sigma = \sigma_i$ and $s = p_i + z_i$. Condition $(\mathbf{T}_2)$ makes the algorithm well-defined when $\sigma^* = -\lambda$, and often allows the algorithm to terminate without the need to compute a zero of $\phi(\sigma)$ to high accuracy.

### *5.3. Properties of an approximate solution*

Given $\delta$ and $\varepsilon$, the convergence and termination conditions given in the previous section provide a nonnegative scalar $\sigma$ and vector $s = p + z$ that satisfy the conditions

$$(B + \sigma T)p = -g \quad \text{and} \quad \|p + z\|_T \leq (1 + \varepsilon)\delta, \tag{5.5}$$

where $B + \sigma T$ is positive semidefinite, and the (possibly zero) vector $z$ is chosen to satisfy the approximate null-vector condition

$$z^T B(\sigma) z \leq \varepsilon(2-\varepsilon)\big(p^T B(\sigma) p + \sigma\delta^2\big). \tag{5.6}$$

More precisely, if $\sigma = 0$, then $z = 0$ and the upper bound on $\|s\|_T$ is $(1-\varepsilon)\delta$; if $\sigma > 0$ and $\|p\|_T \geq (1-\varepsilon)\delta$ then $z = 0$ and $(1-\varepsilon)\delta \leq \|s\|_T \leq (1+\varepsilon)\delta$; finally, if $\sigma > 0$ and $\|p\|_T < (1-\varepsilon)\delta$, then $z \neq 0$ and $\|s\|_T = \delta$. These inequalities imply that the parameter $\nu$ of Algorithm 3.1 has the value $1/(1-\varepsilon)$.

The next lemma shows that any $s$ satisfying the conditions (5.5) and (5.6) also satisfies condition (3.6) for a certain $\tau$ and related trust-region radius $\hat{\delta}$.

**Lemma 5.2.** *Let $\varepsilon$ be a fixed scalar such that $0 < \varepsilon < 1$. Consider any $\sigma \geq 0$ and vector $s = p + z$ satisfying* (5.5) *and* (5.6) *with $B + \sigma T$ positive semidefinite. Then $s$ satisfies*

$$\mathcal{Q}(s) \leq \tau \mathcal{Q}^* \quad \text{and} \quad \|s\|_T \leq \hat{\delta}, \tag{5.7}$$

*where $\tau = \big((1-\varepsilon)/(1+\varepsilon)\big)^2$, $\hat{\delta} = (1+\varepsilon)\delta$ and $\mathcal{Q}^* = \min\{\, \mathcal{Q}(s) : \|s\|_T \leq \hat{\delta}\,\}$.*

*Proof.* See Moré and Sorensen [45] and Gertz [32]. □

This result implies that the approximate solution of the trust-region subproblem satisfies the decrease requirement (3.6) with a slightly larger value of the trust-region radius. Condition (3.6) is simpler to use when proving theoretical results, but conditions (5.5) and (5.6) are more appropriate for the discussion of the algorithm for the trust-region subproblem.

### *5.4. The safeguarded Newton iteration*

If $g \neq 0$, the function $\phi(\sigma)$ is strictly decreasing and strictly convex, and it is straightforward to verify the following results.

**Lemma 5.3.** *Suppose that there is a $\sigma^0 \in (-\lambda, \infty)$ such that $\phi(\sigma^0) = 0$. Let $\mathcal{N}(\sigma_i)$ denote the Newton iterate*

$$\mathcal{N}(\sigma_i) = \sigma_i - \phi(\sigma_i)/\phi'(\sigma_i)$$

*for some $\sigma_i \in (-\lambda, \infty)$. Then*

(a) *if $\sigma_i \neq \sigma^0$, the products $\phi(\sigma_i)(\sigma_i - \sigma^0)$ and $\psi(\sigma_i)(\sigma_i - \sigma^0)$ are negative;*
(b) *if $\sigma_i > \sigma^0$ then $\mathcal{N}(\sigma_i) < \sigma^0$;*
(c) *if $\sigma_i \in (-\lambda, \sigma^0)$ then $\mathcal{N}(\sigma_i) > \sigma_i$ and $\mathcal{N}(\sigma_i) \in (-\lambda, \sigma^0)$; and*
(d) *if $\sigma_i \in (-\lambda, \sigma^0)$ then all subsequent Newton iterates increase monotonically and converge to $\sigma^0$.* □

Since the termination criteria for the Newton iteration are based on the value of $\psi(\sigma)$, we state the next result for both $\phi$ and $\psi$.

**Lemma 5.4.** *Suppose there is no $\sigma^0 \in (-\lambda, \infty)$ such that $\phi(\sigma^0) = 0$. If $\sigma_i > -\lambda$, then $\mathcal{N}(\sigma_i) < -\lambda$ and both $\phi(\sigma_i)$ and $\psi(\sigma_i)$ are negative.* □

**Corollary 5.1.** *If $\sigma^* = 0$ and $\sigma_i > 0$, then $\mathcal{N}(\sigma_i) < 0$ and both $\phi(\sigma_i)$ and $\psi(\sigma_i)$ are negative.* □

The goal of the safeguarding procedure is to define a valid iterate in situations where $\mathcal{N}(\sigma_i) < -\lambda$ or $\mathcal{N}(\sigma_i) < 0$. The safeguarded iteration produces a nested sequence of half-open intervals $\{\mathcal{J}_i\}$ such that $\mathcal{J}_i = (a_i, b_i]$ with $\mathcal{J}_{i+1} \subset \mathcal{J}_i$. If $\mathcal{N}(\sigma_i)$ is not a valid iterate, then $\sigma_{i+1}$ is chosen as a positive weighted average of the endpoints $a_{i+1}$ and $b_{i+1}$ lying in the interior of $\mathcal{J}_{i+1}$.

Lemma 5.3 implies that if $\sigma^* \neq 0$ and $\sigma^* \neq -\lambda$, then there is a nonempty interval $\mathcal{D} = (\max\{\,0, -\lambda\,\}, \sigma^*]$ of desirable starting points for which an unmodified Newton iteration will be well-defined and converge to $\sigma^*$. Suppose that $\mathcal{D}$ is nonempty. Then by design there is an interval $\widehat{\mathcal{D}} \subset \mathcal{D}$ of positive length such that $\widehat{\mathcal{D}} \subset \mathcal{J}_i$ for all $i$. In Theorem 5.1, it is shown that sequences $\{\mathcal{J}_i\}$ and $\{\sigma_i\}$ are generated so that if $\sigma_i \notin \mathcal{D}$, then $\sigma_i \in \mathcal{J}_i$ and $b_{i+2} - a_{i+2} \leq \gamma(b_i - a_i)$ for some $0 < \gamma < 1$. It follows that there is some finite $q$ for which $\sigma_q \in \widehat{\mathcal{D}} \subset \mathcal{D}$. Care must be taken to ensure that the algorithm converges when $\mathcal{D}$ is empty; i.e., when $\sigma^* = 0$ or $\sigma^* = -\lambda$. These cases are the subject of Theorems 5.2 and 5.3 respectively.

The algorithm requires routines $\bar{\lambda}_{\min}(A)$ and $z_{\text{null}}(B, \sigma, T, p, \delta)$. The routine $\bar{\lambda}_{\min}(A)$ computes a lower bound estimate of the least eigenvalue of a symmetric matrix $A$, i.e., $\bar{\lambda}_{\min}(A) \leq \lambda_{\min}(A)$. The routine $z_{\text{null}}(B, \sigma, T, p, \delta)$ computes a scaled approximate null-vector $z$ of $B + \sigma T$. In particular, $z$ is such that $z^T(B + \sigma T)z$ is small and $\|p + z\|_T = \delta$. Then under suitable conditions, if $g \neq 0$ the following algorithm will produce a step $s$ that satisfies condition (5.5).

Algorithm 5.1. Trust-region Subproblem.
Specify constants $0 < \varepsilon < 1$, and $0 < \omega < 1$;
Choose $\sigma_0 \geq 0$;
$\bar{b} \leftarrow 1.05 \times \max\big(1, -\bar{\lambda}_{\min}(\widehat{B})\big)/(1 - \omega)$; $[a_0, b_0] \leftarrow [-1, \max\{\,\sigma_0, \bar{b}\,\}]$;
converged $\leftarrow$ **false**; $i \leftarrow 0$;
**while not** converged **do**
  $\big(\lambda^+, \lambda^-, \lambda^0\big) \leftarrow \text{In}\,\big(K(\sigma_i)\big)$;
  **if** $\lambda^+ < n$ **then**
    $[a_{i+1}, b_{i+1}] \leftarrow [\sigma_i, b_i]$; $\sigma_{i+1} \leftarrow \frac{1}{2}(a_{i+1} + b_{i+1})$;
  **else**
    $p \leftarrow -B(\sigma_i)^{-1} g$;
    **if** $|\psi(\sigma_i)| < \varepsilon$ **or** $\big(\psi(\sigma_i) < \varepsilon$ **and** $\sigma_i = 0\big)$ **then**
      $s \leftarrow p$; converged $\leftarrow$ **true**;
    **else if** $\psi(\sigma_i) \leq -\varepsilon$ **and** $\sigma_i > 0$ **then**
      $z \leftarrow z_{\text{null}}\big(B, \sigma_i, T, p, \delta\big)$;
      **if** $z^T B(\sigma_i) z > \varepsilon(2 - \varepsilon)\big(p^T B(\sigma_i) p + \sigma_i \delta^2\big)$ **then**
        $\bar{a}_i \leftarrow \sigma_i - z^T B(\sigma_i) z / \|z\|_T^2$;
        $[a_{i+1}, b_{i+1}] \leftarrow [\max\{\,a_i, \bar{a}_i\,\}, \sigma_i]$;
      **else**
        $s \leftarrow p + z$; converged $\leftarrow$ **true**;
      **end if**

```
        else
            [a_{i+1}, b_{i+1}] ← [a_i, b_i];
        end if
        if not converged then
            σ̄_{i+1} ← max{ 0, N(σ_i) };
            if σ̄_{i+1} > a_{i+1} then
                σ_{i+1} ← σ̄_{i+1}
            else
                σ_{i+1} ← ωa_{i+1} + (1 − ω)b_{i+1};
            end if
        end if
    end if
    i ← i + 1;
end do
```

As our safeguarding techniques are somewhat different from those used in the Moré-Sorensen algorithm, we briefly describe their convergence properties.

**Lemma 5.5.** *Suppose $g$ is nonzero. Define the interval $\mathcal{D} = (\max\{0, -\lambda\}, \sigma^*]$, where $\mathcal{D}$ is empty if $\sigma^* = 0$ or $\sigma^* = -\lambda$. Let $q$ be the smallest integer for which $\sigma_q \in \mathcal{D}$ with $q = \infty$ if no such iterate exists. Suppose $\mathcal{D} \subset \mathcal{J}_0 = (a_0, b_0]$. Then for $i < q$, it holds that the intervals $\{\mathcal{J}_i\}$ are ordered by inclusion, that $\mathcal{D} \subset \mathcal{J}_i$ and that both $\sigma_i \in \mathcal{J}_i$ and $\sigma^* \in \mathcal{J}_i$. Moreover,*

$$b_{i+2} - a_{i+2} \leq \max\left\{\tfrac{1}{2}, 1 - \omega\right\}(b_i - a_i) \tag{5.8}$$

*for all $i \geq 0$ for which $i + 2 < q$.*

*Proof.* We proceed by induction. By assumption, $\mathcal{D} \subset \mathcal{J}_0$ and $\sigma^* \in \mathcal{J}_0$. The choice of $\sigma_0$ and $\mathcal{J}_0$ immediately gives $\sigma_0 \in (a_0, b_0]$.

Suppose $\mathcal{D} \subset \mathcal{J}_i$, $\sigma_i \in \mathcal{J}_i$ and $\sigma^* \in \mathcal{J}_i$. Three cases are possible. If $\sigma_i \in \mathcal{D}$, then $i \geq q$ and there is nothing to prove. Otherwise, it must hold that either $\sigma_i \leq -\lambda$ or $\sigma_i > \sigma^*$.

If $\sigma_i \leq -\lambda$, then $\sigma_{i+1}$ is chosen by bisection and

$$b_{i+1} - a_{i+1} = \tfrac{1}{2}(b_i - \sigma_i) < \tfrac{1}{2}(b_i - a_i).$$

In this case, $\mathcal{J}_{i+1}$ is updated as $b_{i+1} = b_i$ and $a_{i+1} = \sigma_i \leq -\lambda$, and it follows that $\mathcal{D} \in \mathcal{J}_{i+1} \subset \mathcal{J}_i$, $\sigma_{i+1} \in (a_{i+1}, b_{i+1}]$ and $\sigma^* \in (a_{i+1}, b_{i+1}]$.

If $\sigma_i > \sigma^*$, then by Lemma 5.3, $\psi(\sigma_i)$ is negative, with

$$[a_{i+1}, b_{i+1}] = [\max\{a_i, \bar{a}_i\}, \sigma_i], \tag{5.9}$$

where $\bar{a}_i = \sigma_i - z^T B(\sigma_i) z / \|z\|_T^2$. It is not difficult to see that $\bar{a}_i \leq -\lambda \leq 0$ (see Moré and Sorensen [45] for details). Hence $\mathcal{D} \subset \mathcal{J}_{i+1} \subset \mathcal{J}_i$. Moreover, $\mathcal{N}(\sigma_i) < \sigma^* < \sigma_i \leq b_i$ and so $\bar{\sigma}_{i+1} \leq \sigma_i$. However, in this case the rule

$$\sigma_{i+1} = \begin{cases} \bar{\sigma}_{i+1} & \text{if } \bar{\sigma}_{i+1} > a_i, \\ \omega a_{i+1} + (1 - \omega) b_{i+1} & \text{otherwise,} \end{cases}$$

implies that $\sigma_{i+1} \in (a_{i+1}, b_{i+1}]$. Hence, for $i < q$ we may conclude that the intervals $\{\mathcal{J}_i\}$ are ordered by inclusion, that $\mathcal{D} \subset \mathcal{J}_i$ and that both $\sigma_i \in \mathcal{J}_i$ and $\sigma^* \in \mathcal{J}_i$. It remains to show that the inequality (5.8) holds.

Now consider the length of $\mathcal{J}_{i+2}$. Observe that for any $\ell > 0$, if $\sigma_\ell \leq \sigma^*$ then either $\sigma_\ell \in \mathcal{D}$ or $\sigma_\ell < -\lambda$. If $\sigma_\ell \in \mathcal{D}$ then all subsequent iterates are in $\mathcal{D}$ and the inequality (5.8) is irrelevant. If $\sigma_\ell < -\lambda$ then both $\sigma_{\ell+1}$ and $(a_{\ell+1}, b_{\ell+1}]$ are chosen by bisection and the inequality (5.8) will hold for both iterations $i = \ell$ and $i = \ell - 1$.

Thus, we need only consider the case in which both $\sigma_i > \sigma^*$ and $\sigma_{i+1} > \sigma^*$. If $\sigma_i > \sigma^*$, then two situations are possible. If $\sigma_{i+1} = \max\{\, 0, \mathcal{N}(\sigma_i) \,\}$, then from Lemma 5.3 and the assumption that $\sigma^* > 0$, it follows that $\sigma_{i+1} \leq \sigma^*$. Otherwise, $\sigma_{i+1}$ is defined by the rule $\sigma_{i+1} = \omega a_{i+1} + (1 - \omega) b_{i+1}$. Suppose $\sigma_{i+1} > \sigma^*$. In this case, $\psi(\sigma_{i+1})$ is negative, the interval $\mathcal{J}_{i+2}$ is defined by (5.9) and

$$b_{i+2} - a_{i+2} > \sigma_{i+1} - a_{i+1} = (1 - \omega) b_{i+1} + \omega a_{i+1} - a_{i+1} \geq (1 - \omega)(b_i - a_i).$$

Thus (5.8) holds and the lemma is proved. □

**Theorem 5.1.** *Suppose $g$ is nonzero. Define the interval $\mathcal{D} = (\max\{\, 0, -\lambda \,\}, \sigma^*]$. If $\max\{\, 0, -\lambda \,\} < \sigma^*$ then Algorithm 5.1 will produce an iterate $\sigma_q \in \mathcal{D}$ or terminate under conditions $(\mathbf{T}_1)$ or $(\mathbf{T}_2)$ before that occurs.*

*Proof.* Assume that Algorithm 5.1 does not terminate before producing an iterate $\sigma_q \in \mathcal{D}$.

If $b_0 \geq \sigma^*$, then the conditions of Lemma 5.5 are met. But then, because $\sigma^* > \max\{\, 0, -\lambda \,\}$, the interval $\mathcal{D}$ has a positive length and so the bound (5.8) together with the inclusions $\sigma_0 \in (a_0, b_0]$ and $\mathcal{D} \subset \mathcal{J}_0$ imply that $q$ is finite.

If, on the other hand, $b_0 < \sigma^*$, then either $\sigma_0 \in \mathcal{D}$ or $\sigma_0 \leq -\lambda$. In the latter case, the iterates are chosen by repeatedly bisecting the intervals $\mathcal{J}_i$ until $\sigma_i > -\lambda$, and hence $\sigma_i \in \mathcal{D}$. □

If $\sigma^* = 0$ or $\sigma^* = -\lambda$, then $\mathcal{D}$ is empty and Theorem 5.1 does not apply. The case in which $\sigma^* = 0$ is a desirable special case, because then $s$ is an unmodified Newton iterate for the underlying optimization algorithm. Therefore, Algorithm 5.1 has been designed to favor $\sigma = 0$ as a solution to the subproblem.

**Theorem 5.2.** *If $\sigma^* = 0$, then either $\sigma_0$ or $\sigma_1$ is zero.*

*Proof.* If $\sigma_0 \neq 0$ and $\sigma^* = 0$, then $\sigma_0 > \sigma^* \geq -\lambda$ and so $\bar{\sigma}_1 = \max\{\, 0, \mathcal{N}(\sigma_0) \,\}$ is defined. But by Corollary 5.1, $\mathcal{N}(\sigma_0)$ is negative and hence $\bar{\sigma}_1 = 0$. Furthermore, $a_1 = \max\{\, a_1, \bar{a}_1 \,\}$, where $\bar{a}_1 = \sigma_0 - z^T B(\sigma_i) z / \|z\|_T^2$. As discussed in the proof of Theorem 5.1, $\bar{a}_1 \leq -\lambda$. Therefore $a_{i+1} \leq 0 = \bar{\sigma}_1$, and hence $\sigma_1 = 0$. □

If $\sigma^* = 0 > -\lambda$, then $\sigma = 0$ and $s = p = -B^{-1} g$ satisfy conditions (5.5) and (5.6), and Algorithm 5.1 will terminate. Neither Theorem 5.1 nor Theorem 5.2 implies convergence when $\sigma^* = -\lambda$. Theorem 5.1 cannot be applied because $\mathcal{D}$ is empty. Theorem 5.2 implies that if $\sigma^* = -\lambda = 0$, then either $\sigma_0$ or $\sigma_1$ will be zero. However, if $\lambda = 0$ then $B$ is not invertible and $s$ cannot be defined as $-B^{-1} g$.

**Theorem 5.3.** *If $\sigma^* = -\lambda$ then either* Algorithm 5.1 *finds a point satisfying the convergence criteria in a finite number of iterations, or $\sup_{j \geq i} \sigma_j$ converges to $\sigma^*$ from above.*

*Proof.* Suppose the algorithm does not find a point that satisfies the convergence criteria. The interval $\mathcal{D}$ is empty and so the conditions of Lemma 5.5 are trivially met. We may therefore conclude that $\lim_{i \to \infty} \sigma_i = \sigma^*$. We must now show that $\sigma_i > \sigma^*$ infinitely often. But if $\sigma_i \leq \sigma^* = -\lambda$, subsequent iterates are chosen by bisection until one is greater than $-\lambda$. □

It is important that $\sigma_i > \sigma^*$ holds infinitely often, because $z_{\text{null}}(\cdot)$ is only defined for $\sigma_i > -\lambda$. If the $z_{\text{null}}(\cdot)$ routine can be guaranteed to produces a $z$ satisfying

$$z^T B(\sigma_i) z \leq \varepsilon(2 - \varepsilon)\big(p^T B(\sigma_i) p + \sigma_i \delta^2\big)$$

as $\sigma_i$ converges to $-\lambda$ from above, then Theorem 5.3 guarantees that Algorithm 5.1 must terminate after finitely many iterations.

Moré and Sorensen define a routine $z_{\text{null}}(\cdot)$ that uses the Cholesky factors of $\widehat{B} + \sigma I$ and the condition estimator proposed by Cline, Moler, Stewart and Wilkinson [11]. Since our method does not compute an explicit factorization of $\widehat{B} + \sigma I$, we define $z_{\text{null}}(\cdot)$ using the condition estimator `DLACON` supplied with LAPACK [1]. This routine generates an approximate null vector using Higham's [41] modification of Hager's algorithm [39]. This routine uses matrix-vector products with $(B + \sigma T)^{-1}$, rather than a matrix factorization, to estimate $\|(\widehat{B} + \sigma I)^{-1}\|_1$. By-products of the computation of $\|(\widehat{B} + \sigma I)^{-1}\|_1$ are vectors $v$ and $w$ such that $w = (\widehat{B} + \sigma I)^{-1} v$, $\|v\|_1 = 1$ and

$$\|(\widehat{B} + \sigma I)^{-1} v\|_1 = \|w\|_1 \approx \|(\widehat{B} + \sigma I)^{-1}\|_1 = \max_{\|u\|_1 = 1} \|(\widehat{B} + \sigma I)^{-1} u\|_1.$$

Thus, unless $\|w\| = 0$, the vector $y = w/\|w\|$ is a unit approximate null vector from which we determine an appropriate $z$ such that $\|p + z\|_T = \delta$. The implementation in LAPACK computes at most five matrix-vector products, and so the work associated with Hager's algorithm is proportional to a small multiple of $(m + n)^2$.

The disadvantage of using Hager's algorithm is a theoretical one. Moré and Sorensen are able to show that if $\sigma^* = -\lambda$, and the algorithm of Cline *et al.* is used to find $y$, then $y^T \widehat{B}(\sigma) y \to 0$. We are not aware of an analogous proof for Hager's algorithm, and without such a proof we cannot prove that Algorithm 5.1 converges when $\sigma^* = -\lambda$. However, we note that Moré and Sorensen's proof assumes the use of exact arithmetic and does not consider the instabilities associated with the Cholesky factorization of a near-singular matrix (see Higham [42] for details). Notwithstanding this, we would prefer to have a theoretical guarantee of convergence, but are obliged to use Hager's algorithm for practical reasons.

Another difference between Algorithm 5.1 and the Moré-Sorensen algorithm concerns the definition of $z$ once $y$ has been computed. For every $y$ there are

two choices of $z$, and unless $g^T y = 0$, one value of $z$ gives a negative $g^T z$ and the other gives a positive $g^T z$. We always ensure $g^T z \leq 0$, so that $g^T(p+z) \leq 0$, whereas Moré and Sorensen choose $z$ to minimize $\mathcal{Q}(p+z)$. In theory, either choice is equally valid, but the Moré-Sorensen choice may give $g^T(p+z) > 0$, which would prevent $s = p + z$ from being a descent direction for the line search. (In practice, our choice of sign often gives the Moré-Sorensen direction.)

The last and most significant departure from the Moré-Sorensen algorithm is that we cannot determine a good upper bound on $\lambda_{\min}(B+\sigma T)$ when $B+\sigma T$ is indefinite. During the attempt to compute a Cholesky factorization of an indefinite matrix, it is straightforward to find a direction of negative curvature (and hence an upper bound on $\lambda_{\min}$). It is not clear, however, how to find such an estimate from a general factorization of (2.5). The availability of a good upper bound on $\lambda_{\min}$ can accelerate the convergence of Algorithm 5.1, but does not alter the convergence theory.

## 6. Numerical experiments

In this section we describe some numerical results on the problems in the COPS 2.0 test collection [5,16,17] implemented in the AMPL modeling language [28]. The results were obtained using a MATLAB implementation of the primal-dual trust-region method defined by Algorithms 1.1 and 5.1.

### *6.1. Extensions*

For simplicity, the theoretical results of Section 4 are presented in terms of a single parameter $\mu$ for both the penalty and barrier terms. A trivial modification of the theory allows the use of different parameters—$\mu_{\mathcal{E}}$ for the penalty term and $\mu_{\mathcal{I}}$ for the barrier term. With this modification, the residual function for the perturbed optimality conditions is defined as

$$r(x,y) = \begin{cases} c_i(x) + \mu_{\mathcal{E}} y_i & \text{for } \; i \in \mathcal{E}, \\ c_i(x) y_i - \mu_{\mathcal{I}} & \text{for } \; i \in \mathcal{I} \end{cases}$$

(see the definition (1.3)). Let $c_{\mathcal{E}}$ and $c_{\mathcal{I}}$ denote the components of $c$ corresponding to the index sets $\mathcal{E}$ and $\mathcal{I}$ respectively, with analogous definitions for $y_{\mathcal{E}}$ and $y_{\mathcal{I}}$. Given penalty and barrier parameters $\mu_{\mathcal{E}}$ and $\mu_{\mathcal{I}}$, the inner iterations are terminated when $x$ and $y$ satisfy $t_{\mathcal{E}}\big(x,y,\mu_{\mathcal{E}}\big) \leq \max\{\,\mu_{\mathcal{E}},\mu_{\mathcal{I}}\,\}$ and $t_{\mathcal{I}}\big(x,y,\mu_{\mathcal{I}}\big) \leq \mu_{\mathcal{I}}$, where

$$t_{\mathcal{E}}\big(x,y,\mu_{\mathcal{E}}\big) = \max\big\{\|\nabla f - J^T y\|_\infty,\, \|c_{\mathcal{E}} - \mu_{\mathcal{E}} y_{\mathcal{E}}\|_\infty\big\}, \tag{6.1a}$$

$$\text{and} \qquad t_{\mathcal{I}}\big(x,y,\mu_{\mathcal{I}}\big) = \tau \times \max_{i\in\mathcal{I}}\{\,c_i y_i\,\}, \tag{6.1b}$$

with $\tau \in (0, \frac{1}{2}]$ a parameter with value given below. This test is less stringent than requiring $\|F^{\mu}(x,y)\|_\infty \leq \max\{\,\mu_{\mathcal{E}},\mu_{\mathcal{I}}\,\}$, and we have observed empirically

that when conditions (6.1) are satisfied it is usually beneficial to reduce $\mu_\varepsilon$ and $\mu_\mathcal{I}$ rather than to continue to find a smaller value of $\|F^\mu\|_\infty$.

The initial barrier and penalty parameters are defined as

$$\mu_\varepsilon^{(0)} = 10^{-2} \times \max\{\|\nabla f - J^T y\|_\infty,\, \|c_\varepsilon\|_\infty,\, 10^{-10}\},$$
$$\text{and} \qquad \mu_\mathcal{I}^{(0)} = \max\{\|F^\infty\|_\infty, 10^{-8}\},$$

where all quantities are evaluated at the primal-dual starting point $\big(x^{(0)}, y^{(0)}\big)$ (see Section 6.4 below). The updating rules for these parameters involve the quantity

$$\xi_k = \max\{\|\nabla f - J^T y\|_\infty,\, \|c_\varepsilon + \mu_\varepsilon^{(k)} y_\varepsilon\|_\infty\},$$

where, in this case, all quantities are evaluated at the $k$th outer iterate $\big(x^{(k)}, y^{(k)}\big)$. The following rule for updating $\mu_\varepsilon$ is superlinear. Given an initial scale factor $\beta_0$ $(0 < \beta_0 < 1)$, we define

$$\mu_\varepsilon^{(k+1)} = \max\{\beta_k \times \min\{\,\mu_\varepsilon^{(k)},\, \xi_k\,\},\, 10^{-8}\} \quad \text{and} \quad \beta_{k+1} = \beta_k^{3/2}.$$

The rule for updating $\mu_\mathcal{I}^{(k)}$ is more conservative. We choose

$$\mu_\mathcal{I}^{(k+1)} = \max\Big\{\frac{1}{\tau}(1+\tau) \times \min\big\{\mu_\mathcal{I}^{(k)},\, \max_{i\in\mathcal{I}}\{\,c_i y_i\,\}\big\},\, \frac{\xi_k}{100},\, 10^{-8}\Big\},$$

where $\tau$ is the parameter used to terminate the inner iterations (see, e.g., (6.1b)). The values $\beta_0 = .81$ and $\tau = \frac{1}{9}$ are used in all the numerical experiments.

The generic termination criterion for Algorithm 1.1 must also be modified to allow for poorly scaled problems. The outer iterations are terminated at the first primal-dual point $(x, y)$ that satisfies the inequalities:

$$\|\nabla f - J^T y\|_\infty \le \texttt{tol} \times (1 + \|y\|_\infty),$$
$$\|c_\varepsilon\|_\infty \le \texttt{tol} \times (1 + \|x\|_\infty),$$
$$\text{and} \qquad \max_{i\in\mathcal{I}}\{\,c_i y_i\,\} \le \texttt{tol} \times (1 + \|y_\mathcal{I}\|_\infty),$$

where $\texttt{tol} = 10^{-6}$ in the numerical experiments.

### *6.2. The* MATLAB *implementation*

The MATLAB implementation is part of a general-purpose optimization testing environment intended to support the rapid development of algorithms independently of the problem format. In this environment, all problems are converted to the generic form (NP) and all derivative matrices as treated as dense. The MATLAB interior-point solver forms a preliminary implementation of the C$^{++}$ program IOTR (**I**nterior methods for **O**ptimization using **T**rust **R**egions) being developed jointly at the University of California, San Diego and the University of Wisconsin, Madison. The algorithm fully implements Algorithm 5.1 for the solution of the trust-region subproblem. The code uses the dense linear algebra subroutines

DSYTF2, DSYTRS and DLACON from LAPACK [1]. The routines DSYTF2 and DSYTRS factor and solve the symmetric indefinite system using Bunch-Kaufman symmetric pivoting. Subroutine DLACON is used to compute the approximate null-vector in the hard case.

In AMPL, all constraints are formulated as $l_j \leq \phi_j(x) \leq u_j$, where $l_j$ and $u_j$ are constants such that $l_j = u_j$ for an equality constraint, and $|l_j|$ or $|u_j|$ is infinite when no bound is present. The function $\phi_j(x)$ is either a simple variable $\phi_j(x) = x_j$, or a general linear or nonlinear function. The format of problem (NP) makes no distinction between simple bounds and general constraints. Moreover, inequality constraints $l_j \leq \phi_j(x) \leq u_j$ with finite upper and lower bounds are treated as two constraints $\phi_j(x) \geq l_j$ and $-\phi_j(x) \geq -u_j$. Given that the code treats all Jacobians and Hessians as dense matrices, these features limit the dimension of the KKT system that may be solved using dense factorization techniques and hence limit the size of problem that may be attempted in the MATLAB environment. (In any production code the computation would be arranged so that the KKT system seen by the solver would be of the dimension $n + m$ where $n$ is the number of variables $x_j$ with $l_j \neq u_j$ and $m$ is the number of terms $l_j \leq \phi_j(x) \leq u_j$ involving general $\phi_j(x)$.)

### *6.3. Properties of the* COPS *problems*

The dimension of a particular "case" or "instance" of a COPS problem is determined by one or more parameters assigned in its AMPL data file. For each of the 17 COPS problems, Dolan and Moré [16] give the results of several optimization algorithms on a range of cases obtained by varying only one of the parameters. Here we consider one problem from each Dolan-Moré selection. In each case, this problem was the largest problem that could be solved in reasonable computation time, subject to the resource limitations imposed by the form of the MATLAB implementation.

Table 6.1 gives the details of the 17 COPS problems used in the numerical experiments. The first three columns give the problem number, name and value of the problem-size parameter (listed in the notation of [16]). The fourth column describes the problem type, where "nc" indicates a nonlinear objective with bounds and general nonlinear constraints (some of which may be linear), and "bc" indicates a nonlinear objective with simple upper and lower bound constraints only. The last four columns give the dimension of the problem, as determined by the parameter given in the "Problem/size" column. The "Vars." column gives the total number of variables, including fixed variables. The column "Constraints/total" gives the number of general constraints of the form $l_j \leq \phi_j(x) \leq u_j$. The columns "Constraints/inEq." and "Constraints/eqn" give the number of general inequalities (i.e., terms with $l_j \neq u_j$) and general equalities (terms with $l_j = u_j$) respectively.

By default, when generating a problem instance from a model file, AMPL uses a so-called "presolve" in an attempt to tighten the upper and lower bounds and eliminate as many bounds and linear constraints as possible. Generating an

**Table 6.1.** COPS problems in AMPL format with default presolve.

| Problem | | | | Vars. | Constraints | | |
|---|---|---|---|---|---|---|---|
| No. | Name | Size | Type | | InEq. | Eq. | Total |
| 1 | *Polygon* | $n_v = 50$ | nc | 98 | 1273 | 0 | 1273 |
| 2 | *Elec* | $n_p = 200$ | nc | 600 | 0 | 200 | 200 |
| 3 | *Chain* | $n_h = 400$ | nc | 800 | 0 | 401 | 401 |
| 4 | *Camshape* | $n = 200$ | nc | 200 | 400 | 0 | 400 |
| 5 | *Pinene* | $n_h = 50$ | nc | 1000 | 0 | 995 | 995 |
| 6 | *Marine* | $n_h = 25$ | nc | 615 | 0 | 592 | 592 |
| 7 | *Channel* | $n_h = 100$ | nc | 900 | 0 | 300 | 300 |
| 8 | *Robot* | $n_h = 50$ | nc | 449 | 0 | 301 | 301 |
| 9 | *Steering* | $n_h = 100$ | nc | 500 | 0 | 401 | 401 |
| 10 | *Rocket* | $n_h = 100$ | nc | 401 | 0 | 300 | 300 |
| 11 | *Glider* | $n_h = 100$ | nc | 499 | 0 | 400 | 400 |
| 12 | *Gasoil* | $n_h = 50$ | nc | 501 | 0 | 498 | 498 |
| 13 | *Methanol* | $n_h = 50$ | nc | 602 | 0 | 597 | 597 |
| 14 | *Catmix* | $n_h = 200$ | nc | 601 | 0 | 400 | 400 |
| 15 | *Torsion* | $n_y = 25$ | bc | 1250 | 0 | 0 | 0 |
| 16 | *Bearing* | $n_y = 25$ | bc | 1250 | 0 | 0 | 0 |
| 17 | *Minsurf* | $n_y = 25$ | bc | 1250 | 0 | 0 | 0 |

instance of an AMPL problem *without* a presolve provides an additional problem for testing. Table 6.2 gives the details of the 17 COPS problems generated without a presolve (invoked using the AMPL command "`option presolve 0`").

**Table 6.2.** COPS problems in AMPL format with no presolve.

| Problem | | | | Vars. | Constraints | | |
|---|---|---|---|---|---|---|---|
| No. | Name | Size | Type | | InEq. | Eq. | Total |
| 1 | *Polygon* | $n_v = 50$ | nc | 100 | 1374 | 2 | 1376 |
| 2 | *Elec* | $n_p = 200$ | nc | 600 | 0 | 200 | 200 |
| 3 | *Chain* | $n_h = 400$ | nc | 802 | 0 | 403 | 403 |
| 4 | *Camshape* | $n = 200$ | nc | 200 | 403 | 0 | 403 |
| 5 | *Pinene* | $n_h = 50$ | nc | 1005 | 5 | 1000 | 1005 |
| 6 | *Marine* | $n_h = 25$ | nc | 615 | 15 | 592 | 607 |
| 7 | *Channel* | $n_h = 100$ | nc | 800 | 0 | 800 | 800 |
| 8 | *Robot* | $n_h = 50$ | nc | 461 | 0 | 313 | 313 |
| 9 | *Steering* | $n_h = 100$ | nc | 507 | 102 | 408 | 510 |
| 10 | *Rocket* | $n_h = 100$ | nc | 405 | 304 | 304 | 608 |
| 11 | *Glider* | $n_h = 100$ | nc | 506 | 304 | 407 | 711 |
| 12 | *Gasoil* | $n_h = 50$ | nc | 503 | 3 | 500 | 503 |
| 13 | *Methanol* | $n_h = 50$ | nc | 605 | 5 | 600 | 605 |
| 14 | *Catmix* | $n_h = 200$ | nc | 603 | 0 | 402 | 402 |
| 15 | *Torsion* | $n_y = 25$ | lc | 1404 | 1250 | 154 | 1404 |
| 16 | *Bearing* | $n_y = 25$ | lc | 1404 | 0 | 158 | 158 |
| 17 | *Minsurf* | $n_y = 25$ | lc | 1404 | 1824 | 158 | 1982 |

The only problem unaltered by the presolve is *Elec*. The others have some substantial differences in formulation. For example, without a presolve, AMPL

formulates the bound constraints of problems *Torsion*, *Bearing* and *Minsurf* as *general* linear constraints (marked as "lc" in the "Problem/type" column). Such differences in formulation can have a significant effect on performance because the code uses a different technique to obtain feasibility for simple bound and general constraints (see Section 6.4).

### *6.4. Starting points*

In all the numerical experiments, the initial values for the dual variables were $y_{\mathcal{E}}^{(0)} = 0$ and $y_{\mathcal{I}}^{(0)} = 1$. For the primal variables, the standard starting point for each COPS model was used whenever possible. However, in some cases it is necessary to redefine some components of the standard start so that the initial point is sufficiently interior with respect to the simple bounds. The precise definition of "sufficiently interior" depends on a positive preassigned scalar $\epsilon$ ($\epsilon = 10^{-1}$ in our tests). Given the COPS standard start $x^C$, the components of the initial primal iterate are defined such that

$$x_j^{(0)} = \min\left\{ \max\{x_j^C, l_j + \delta l_j\},\ u_j - \delta u_j \right\}, \tag{6.2}$$

where $\delta l_j$ and $\delta u_j$ define strictly positive perturbations of the lower and upper bounds. If an upper or lower bound is infinite, then its corresponding perturbation is zero. If both bounds are finite, then $\delta l_j = \epsilon(u_j - l_j)$, otherwise, if $|l_j| < \infty$ and there is no upper bound, then $\delta l_j = \epsilon(1 + |l_j|)$. The value of $\delta u_j$ is defined in a similar way.

If the resulting $x_j^{(0)}$ is not sufficiently interior for a general constraint $l_i \leq \phi_i(x) \leq u_i$, then a slack variable $s_i$ is used to convert the inequalities into the equality $\phi_i(x) - s_i = 0$ and simple bound constraints $l_i \leq s_i \leq u_i$. This transformation is used for all inequality constraints for which

$$\phi_i(x^{(0)}) \neq \min\left\{ \max\{\phi_i(x^{(0)}), l_j + \delta l_j\},\ u_j - \delta u_j) \right\},$$

where $\delta l_j$ and $\delta u_j$ are defined as in the case of simple bounds. The slack is initialized at the value: $\min\left\{ \max\{\phi_i(x^{(0)}), l_j + \delta l_j\},\ u_j - \delta u_j \right\}$.

Tables 6.3 and 6.4 give the problem statistics for the COPS problems written in the form (NP). The first two columns give the problem number and name, as in Table 6.1. The columns "$n$", "$|\mathcal{E}|$" and "$|\mathcal{I}|$" list the number of variables and constraints associated with problem (NP). The value of $|\mathcal{E}|+|\mathcal{I}|$ is the number of distinct finite bounds $l_j$ and $u_j$ associated with the constraints $l_j \leq \phi_j(x) \leq u_j$, plus the number of added slack variables. The entries in the $n$ column reflect the number of variables from Table 6.1 plus the number of added slack variables. The column marked "KKT" gives the dimension of the dense system (2.5) factored and solved at each inner iteration. The last two columns give the number of added slack variables and the number of components of the COPS standard start that were moved inside their upper and lower bounds.

**Table 6.3.** COPS problem statistics in format (NP).

| Problem | | | | | | Modifications | |
|---|---|---|---|---|---|---|---|
| No. | Name | $n$ | $\|\mathcal{E}\|$ | $\|\mathcal{I}\|$ | KKT | Slacks | Infs. $x_j$ |
| 1 | *Polygon* | 316 | 218 | 1469 | 2003 | 218 | -- |
| 2 | *Elec* | 600 | 200 | 0 | 800 | -- | -- |
| 3 | *Chain* | 800 | 401 | 0 | 1201 | -- | -- |
| 4 | *Camshape* | 204 | 4 | 998 | 1206 | 4 | 2 |
| 5 | *Pinene* | 1000 | 995 | 5 | 2000 | -- | 5 |
| 6 | *Marine* | 615 | 592 | 15 | 1222 | -- | 15 |
| 7 | *Channel* | 900 | 300 | 0 | 1200 | -- | -- |
| 8 | *Robot* | 449 | 301 | 601 | 1351 | -- | -- |
| 9 | *Steering* | 500 | 401 | 203 | 1104 | -- | -- |
| 10 | *Rocket* | 401 | 300 | 601 | 1302 | -- | 101 |
| 11 | *Glider* | 499 | 400 | 402 | 1301 | -- | -- |
| 12 | *Gasoil* | 501 | 498 | 3 | 1002 | -- | 3 |
| 13 | *Methanol* | 602 | 597 | 5 | 1204 | -- | -- |
| 14 | *Catmix* | 601 | 400 | 402 | 1403 | -- | 201 |
| 15 | *Torsion* | 1250 | 0 | 2500 | 3750 | -- | -- |
| 16 | *Bearing* | 1250 | 0 | 1250 | 2500 | -- | 625 |
| 17 | *Minsurf* | 1250 | 0 | 1250 | 2500 | -- | 420 |

**Table 6.4.** COPS problem statistics in format (NP). `Presolve = 0`.

| Problem | | | | | | Modifications | |
|---|---|---|---|---|---|---|---|
| No. | Name | $n$ | $\|\mathcal{E}\|$ | $\|\mathcal{I}\|$ | KKT | Slacks | Infs. $x_j$ |
| 1 | *Polygon* | 321 | 223 | 1474 | 2018 | 221 | -- |
| 2 | *Elec* | 600 | 200 | 0 | 800 | -- | -- |
| 3 | *Chain* | 802 | 403 | 0 | 1205 | -- | -- |
| 4 | *Camshape* | 204 | 4 | 1004 | 1212 | 4 | -- |
| 5 | *Pinene* | 1010 | 1005 | 5 | 2020 | 5 | -- |
| 6 | *Marine* | 630 | 607 | 15 | 1252 | 15 | -- |
| 7 | *Channel* | 800 | 800 | 0 | 1600 | -- | -- |
| 8 | *Robot* | 461 | 313 | 613 | 1387 | -- | -- |
| 9 | *Steering* | 507 | 408 | 203 | 1318 | -- | -- |
| 10 | *Rocket* | 510 | 409 | 607 | 1526 | 105 | -- |
| 11 | *Glider* | 507 | 408 | 405 | 1320 | 1 | -- |
| 12 | *Gasoil* | 506 | 503 | 3 | 1012 | 3 | -- |
| 13 | *Methanol* | 605 | 600 | 5 | 1210 | -- | -- |
| 14 | *Catmix* | 603 | 402 | 402 | 1407 | -- | 201 |
| 15 | *Torsion* | 1404 | 154 | 2500 | 4058 | -- | -- |
| 16 | *Bearing* | 1404 | 158 | 1404 | 2966 | -- | 729 |
| 17 | *Minsurf* | 1878 | 632 | 1824 | 4334 | 474 | -- |

### *6.5. Numerical results*

Table 6.5 gives the results of applying the algorithm to the COPS problems defined in Tables 6.3–6.4. The values of the preassigned parameters used for Algorithm 1.1 were $\eta_1 = 10^{-1}$, $\eta_2 = \frac{1}{4}$, $\gamma_2 = \frac{1}{2}$, $\gamma_3 = 2$, and $\nu = 1$. The parameters used for Algorithm 5.1 were $\varepsilon = 10^{-1}$, and $\omega = .99$. The termination criteria for the inner and outer iterations were defined with the parameters $\tau = \frac{1}{9}$ and $\texttt{tol} = 10^{-6}$.

For each problem we list the total number of function evaluations, the average number of KKT factorizations per inner iteration, and the final value of the objective function. When the problems are generated with the presolve phase, IOTR can solve 14 problems, at the total cost of 3515 function evaluations. With the given parameter settings, the algorithm was unable to solve *Pinene*, *Marine*, and *Glider* within an allotted 1000 function evaluations. When no presolve is used, IOTR can solve 15 problems in a total of 3990 function evaluations. In this case, IOTR was unable to solve *Channel* and *Glider* within an allotted 1000 function evaluations.

The average number of factorizations per inner iteration is a measure of the efficiency of the Newton method for the trust-region parameter. The number varies between 1.3 and 4.0 over all the problems, with an average value of 2. Given the remarks of Section 5 we would expect this average to be more than the average of 1.5–2 factorizations per iteration reported by Moré and Sorensen [45] for unconstrained problems. Note that our results are comparable to those of Moré and Sorensen for the bound-constraint versions of the problems *Torsion*, *Bearing* and *Minsurf*.

**Table 6.5.** Performance on the COPS problems.

| Problem | | Presolve | | | No Presolve | | |
|---|---|---|---|---|---|---|---|
| No. | Name | Fcns | Factrs | Final obj. | Fcns | Factrs | Final obj. |
| 1 | *Polygon* | 355 | 2.3 | 1.3263e-05 | 632 | 2.2 | 1.1026e-05 |
| 2 | *Elec* | 321 | 4.0 | 1.8439e+04 | 321 | 4.0 | 1.8439e+04 |
| 3 | *Chain* | 214 | 2.1 | 5.0677e+00 | 214 | 2.1 | 5.0682e+00 |
| 4 | *Camshape* | 81 | 1.9 | 4.0700e+00 | 121 | 2.4 | 4.0700e+00 |
| 5 | *Pinene* | --- | --- | --- | 26 | 1.8 | 1.9621e+01 |
| 6 | *Marine* | 47 | 1.6 | 1.9752e+07 | 141 | 2.7 | 1.9749e+07 |
| 7 | *Channel* | 332 | 4.0 | 4.2132e+04 | --- | --- | --- |
| 8 | *Robot* | 609 | 2.4 | 9.1469e+00 | 171 | 3.0 | 9.1469e+00 |
| 9 | *Steering* | 151 | 2.0 | 5.5460e-01 | 123 | 1.9 | 5.5460e-01 |
| 10 | *Rocket* | --- | --- | --- | 825 | 1.4 | 1.0003e+00 |
| 11 | *Glider* | --- | --- | --- | --- | --- | --- |
| 12 | *Gasoil* | 354 | 2.7 | 5.2364e-03 | 395 | 2.5 | 5.2366e-03 |
| 13 | *Methanol* | 949 | 2.1 | 9.0223e-03 | 897 | 2.1 | 9.0224e-03 |
| 14 | *Catmix* | 19 | 1.6 | -4.8160e-02 | 19 | 1.6 | -4.8161e-02 |
| 15 | *Torsion* | 15 | 1.3 | -4.1745e-01 | 32 | 1.7 | -4.1730e-01 |
| 16 | *Bearing* | 19 | 1.6 | -1.5396e-01 | 14 | 2.2 | 2.6938e-02 |
| 17 | *Minsurf* | 49 | 2.0 | 2.5196e+00 | 59 | 1.9 | 2.5195e+00 |

Some care is needed when interpreting the results of Table 6.5. First, many of the problems are not convex, and the number of function evaluations can vary considerably depending on which local solution is found. In particular, *Polygon*, *Elec* and *Channel* have local solutions (see [16]).

Second, it should be emphasized that although the algorithm was unable to solve every problem with the default parameter settings, successful runs were possible with other sensible choices. For example, if the parameter $\epsilon$ used for the feasible perturbation (6.2) is reduced from $10^{-1}$ to $10^{-2}$, then *Glider* converges in

375 evaluations with a presolve and 952 evaluations without a presolve. The same value of $\epsilon$ allows a presolved *Pinene* to converge in 76 function evaluations. These results suggests that more study of the choice of the initial feasible perturbation is needed.

Overall, given the difficulty of the COPS problems, the somewhat unsophisticated choices for the penalty and barrier parameters, and the simple initial values of the primal-dual variables, the numerical experiments indicate that the proposed algorithm forms the basis for a viable algorithm for computing second-order solutions for large-scale nonlinear optimization problems. In particular, it seems likely that the efficiency and reliability exhibited in Table 6.5 can be improved considerably. (For more sophisticated choices for the barrier parameter and initial dual variables, see, e.g., Gay, Overton and Wright [31], and Moguerza and Prieto [43].)

## 7. Summary

We have formulated and analyzed a primal-dual trust-region interior algorithm for large-scale nonlinear optimization. The main idea of the method is a simple one—to use a trust-region method to minimize an unconstrained function that involves both the primal and dual variables. This is a different approach from that taken in other trust-region interior methods and it has the benefit that the constituent linear systems have the same structure as the conventional primal-dual system. No special factorization need be used for the nonconvex case and hence the system can be solved by a range of efficient off-the-shelf direct solvers, or by methods tailored to particular applications. The algorithm appears to be effective in maintaining the quick convergence and stability of trust-region methods, while significantly decreasing the average cost per iteration. It has been shown that under suitable conditions, the sequence of inner iterations converges to a point that satisfies the second-order necessary conditions for a minimizer of the augmented penalty-barrier function. Finally, we have performed some preliminary numerical testing on some medium-scale problems from the nontrivial COPS test set. The numerical experiments indicate that the proposed algorithm forms the basis for a viable algorithm for computing second-order solutions for large-scale nonlinear optimization problems. In addition, the results serve as a useful predictor of how the method will behave on larger problems. The convergence of the outer iterations and more extensive tests on large-scale problems will be the subject of a separate paper.

*Acknowledgements.* The research contained in this paper was supported National Science Foundation grants DMS-9973276 and ACI-0082100. We are grateful to Stephen J. Wright, Joshua Griffin and the referees for their careful reading of the manuscript and a number of helpful suggestions.

## References

1. E. ANDERSON, Z. BAI, C. BISCHOF, S. BLACKFORD, J. DEMMEL, J. DONGARRA,

J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, third ed., 1999.
2. M. Argaez and R. A. Tapia, *On the global convergence of a modified augmented Lagrangian linesearch interior point Newton method for nonlinear programming*, Technical Report 95-38, Dept of Computational and Applied Mathematics, Rice University, Houston, TX, 1995. Revised February 1997.
3. H. Y. Benson, D. F. Shanno, and R. J. Vanderbei, *Interior-point methods for nonconvex nonlinear programming: Filter methods and merit functions*, Report ORFE-00-06, Department of Operations Research and Financial Engineering, Princeton University, 2000.
4. ———, *Interior-point methods for nonconvex nonlinear programming: Jamming and comparative numerical testing*, Report ORFE-00-02, Department of Operations Research and Financial Engineering, Princeton University, 2000.
5. A. Bondarenko, D. Bortz, and J. J. Moré, *COPS: Large-scale nonlinearly constrained optimization problems*, Technical Report ANL/MCS-TM-237, Mathematics and Computer Science division, Argonne National Laboratory, Argonne, IL, 1998. Revised October 1999.
6. J. R. Bunch and L. Kaufman, *Some stable methods for calculating inertia and solving symmetric linear systems*, Math. Comput., 31 (1977), pp. 163–179.
7. J. R. Bunch and B. N. Parlett, *Direct methods for solving symmetric indefinite systems of linear equations*, SIAM J. Numer. Anal., 8 (1971), pp. 639–655.
8. R. H. Byrd, *Robust trust-region methods for constrained optimization.* Paper presented at the SIAM Conference on Optimization, Houston, TX, 1987.
9. R. H. Byrd, J. C. Gilbert, and J. Nocedal, *A trust region method based on interior point techniques for nonlinear programming*, Math. Program., 89 (2000), pp. 149–185.
10. R. H. Byrd, M. E. Hribar, and J. Nocedal, *An interior point algorithm for large-scale nonlinear programming*, SIAM J. Optim., 9 (1999), pp. 877–900.
11. A. K. Cline, C. B. Moler, G. W. Stewart, and J. H. Wilkinson, *An estimate for the condition number of a matrix*, SIAM J. Numer. Anal., 16 (1979), pp. 368–375.
12. A. R. Conn, N. I. M. Gould, D. Orban, and P. L. Toint, *A primal-dual trust-region algorithm for non-convex nonlinear programming*, Math. Program., 87 (2000), pp. 215–249. Studies in algorithmic optimization.
13. A. R. Conn, N. I. M. Gould, and P. L. Toint, *A primal-dual algorithm for minimizing a non-convex function subject to bound and linear equality constraints*, in Nonlinear optimization and related topics (Erice, 1998), Kluwer Acad. Publ., Dordrecht, 2000, pp. 15–49.
14. J. E. Dennis, Jr., M. Heinkenschloss, and L. N. Vicente, *Trust-region interior-point SQP algorithms for a class of nonlinear programming problems*, SIAM J. Control Optim., 36 (1998), pp. 1750–1794.
15. J. E. Dennis Jr. and J. J. Moré, *A characterization of superlinear convergence and its application to quasi-Newton methods*, Math. Comput., 28 (1974), pp. 549–560.
16. E. D. Dolan and J. J. Moré, *Benchmarking optimization software with cops*, Technical Memorandum ANL/MCS-TM-246, Argonne National Laboratory, 2000.
17. ———, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213.
18. A. S. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang, *On the formulation and theory of the Newton interior-point method for nonlinear programming*, J. Optim. Theory Appl., 89 (1996), pp. 507–541.
19. A. V. Fiacco and G. P. McCormick, *Nonlinear Programming*, Classics in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second ed., 1990. Reprint of the 1968 original.
20. R. Fletcher, *Practical Methods of Optimization*, John Wiley and Sons, Chichester and New York, second ed., 1987.
21. R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Math. Program., 91 (2002), pp. 239–269.
22. A. Forsgren, *Inertia-controlling factorizations for optimization algorithms*, Appl. Num. Math., 43 (2002), pp. 91–107.
23. A. Forsgren and P. E. Gill, *Primal-dual interior methods for nonconvex nonlinear programming*, SIAM J. Optim., 8 (1998), pp. 1132–1152.
24. A. Forsgren, P. E. Gill, and W. Murray, *Computing modified Newton directions using a partial Cholesky factorization*, SIAM J. Sci. Comput., 16 (1995), pp. 139–150.

25. A. Forsgren, P. E. Gill, and J. R. Shinnerl, *Stability of symmetric ill-conditioned systems arising in interior methods for constrained optimization*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 187–211.
26. A. Forsgren, P. E. Gill, and M. H. Wright, *Interior methods for nonlinear optimization*, SIAM Rev., 44 (2002), pp. 525–597.
27. A. Forsgren and W. Murray, *Newton methods for large-scale linear equality-constrained minimization*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 560–587.
28. R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, Brooks/Cole—Thomson Learning, Pacific Grove, USA, 2003.
29. R. Fourer and S. Mehrotra, *Solving symmetric indefinite systems in an interior-point method for linear programming*, Math. Program., 62 (1993), pp. 15–39.
30. D. M. Gay, *Computing optimal locally constrained steps*, SIAM J. Sci. and Statist. Comput., 2 (1981), pp. 186–197.
31. D. M. Gay, M. L. Overton, and M. H. Wright, *A primal-dual interior method for nonconvex nonlinear programming*, in Advances in Nonlinear Programming (Beijing, 1996), Y. Yuan, ed., Kluwer Acad. Publ., Dordrecht, 1998, pp. 31–56.
32. E. M. Gertz, *Combination Trust-Region Line-Search Methods for Unconstrained Optimization*, PhD thesis, Department of Mathematics, University of California, San Diego, 1999.
33. E. M. Gertz, P. E. Gill, and J. Griffin, *IOTR: An interior point method for optimization using trust regions*, Numerical Analysis Report, University of California, San Diego, 2002. to appear.
34. P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders, *Solving reduced KKT systems in barrier methods for linear and quadratic programming*, Report SOL 91-7, Department of Operations Research, Stanford University, Stanford, CA, 1991.
35. ———, *Solving reduced KKT systems in barrier methods for linear programming*, in Numerical analysis 1993 (Dundee, 1993), G. A. Watson and D. Griffiths, eds., vol. 303 of Pitman Res. Notes Math. Ser., Longman Sci. Tech., Harlow, UK, 1994, pp. 89–104.
36. P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London and New York, 1981.
37. N. I. M. Gould, *On the accurate determination of search directions for simple differentiable penalty functions*, IMA J. Numer. Anal., 6 (1986), pp. 357–372.
38. N. I. M. Gould, D. Orban, A. Sartenaer, and P. L. Toint, *Superlinear convergence of primal-dual interior point algorithms for nonlinear programming*, SIAM J. Optim., 11 (2001), pp. 974–1002.
39. W. W. Hager, *Condition estimates*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 311–316.
40. M. D. Hebden, *An algorithm for minimization using exact second derivatives*, Tech. Report T.P. 515, Atomic Energy Research Establishment, Harwell, England, 1973.
41. N. Higham, *FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation*, ACM Trans. Math. Software, 14 (1988), pp. 381–396.
42. N. J. Higham, *Analysis of the Cholesky decomposition of a semi-definite matrix*, in Reliable Numerical Computation, M. G. Cox and S. Hammarling, eds., Oxford University Press, 1990, pp. 161–185.
43. J. M. Moguerza and F. J. Prieto, *An augmented Lagrangian interior-point method using directions of negative curvature*, Math. Program., 95 (2003), pp. 573–616.
44. J. J. Moré and D. C. Sorensen, *On the use of directions of negative curvature in a modified Newton method*, Math. Program., 16 (1979), pp. 1–20.
45. ———, *Computing a trust region step*, SIAM J. Sci. and Statist. Comput., 4 (1983), pp. 553–572.
46. J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.
47. E. O. Omojokun, *Trust Region Algorithms for Nonlinear Equality and Inequality Constraints*, PhD thesis, Department of Computer Science, University of Colorado, Boulder, 1989.
48. D. B. Ponceleón, *Barrier methods for large-scale quadratic programming*, Report SOL 91-2, Department of Operations Research, Stanford University, Stanford, CA, 1991. PhD thesis.
49. M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., London and New York, 1969, Academic Press, pp. 283–298.

50. ———, *A new algorithm for unconstrained optimization*, in Nonlinear Programming (Proc. Sympos., Univ. of Wisconsin, Madison, Wis., 1970), Academic Press, New York, 1970, pp. 31–65.
51. ———, *Convergence properties of a class of minimization algorithms*, in Nonlinear programming, 2 (Proc. Sympos. Special Interest Group on Math. Programming, Univ. Wisconsin, Madison, Wis., 1974), Academic Press, New York, 1974, pp. 1–27.
52. C. H. Reinsch, *Smoothing by spline functions II*, Numer. Math., 16 (1971), pp. 451–454.
53. D. F. Shanno and R. J. Vanderbei, *Interior-point methods for nonconvex nonlinear programming: orderings and higher-order methods*, Math. Program., 87 (2000), pp. 303–316.
54. D. C. Sorensen, *Newton's method with a model trust region modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.
55. S. W. Thomas, *Sequential estimation techniques for quasi-Newton algorithms*, PhD thesis, Cornell University, 1975.
56. M. Ulbrich, S. Ulbrich, and L. Vicente, *A globally convergent primal-dual interior-point filter method for nonlinear programming*. Manuscript, 2000.
57. R. J. Vanderbei and D. F. Shanno, *An interior-point algorithm for nonconvex nonlinear programming*, Comput. Optim. Appl., 13 (1999), pp. 231–252.
58. A. Wächter and L. T. Biegler, *Global and local convergence for a class of interior point methods for nonlinear programming*, Tech. Report B-01-09, CAPD, Department of Chemical Engineering, Carnegie Mellon University, 2001.
59. R. A. Waltz and J. Nocedal, *KNITRO 2.0 User's Manual*, Report OTC 2002/02, Optimization Technology Center, Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208, 2002.
60. M. H. Wright, *Interior methods for constrained optimization*, in Acta Numerica, 1992, Cambridge University Press, New York, USA, 1992, pp. 341–407.
61. ———, *Ill-conditioning and computational error in interior methods for nonlinear programming*, SIAM J. Optim., 9 (1998), pp. 84–111.
62. S. J. Wright, *Stability of linear equations solvers in interior-point methods*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1287–1307.
63. ———, *Primal-dual interior-point methods*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.
64. S. J. Wright and D. Orban, *Properties of the log-barrier function on degenerate nonlinear programs*, Preprint ANL/MCS-P772-0799, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill., July 1999.
65. H. Yabe, H. Yamashita, and T. Tanabe, *A primal-dual interior point method for large-scale nonlinear optimization problems*, Sūrikaisekikenkyūsho Kōkyūroku, (1998), pp. 204–211. Research on algorithms for numerical computing (Japanese) (Kyoto, 1997).
66. Y. Zhang, R. Tapia, and F. Potra, *On the superlinear convergence of interior-point algorithms for a general class of problems*, SIAM J. Optim., 3 (1993), pp. 413–422.
67. Y. Zhang and R. A. Tapia, *A superlinearly convergent polynomial primal-dual interior-point algorithm for linear programming*, SIAM J. Optim., 3 (1993), pp. 118–133.
68. Y. Zhang, R. A. Tapia, and J. Dennis, J. E., *On the superlinear and quadratic convergence of primal-dual interior point linear programming algorithms*, SIAM J. Optim., 2 (1992), pp. 304–324.