A Primal-Dual Trust-Region Algorithm for Minimizing a Non-convex Function Subject to General Inequality and Linear Equality Constraints

by A. R. Conn[1], N. I. M. Gould[2],
D. Orban[3] and Ph. L. Toint[4]

Report 99/04 16 August 1999

[1] IBM T.J. Watson Research Center,
P.O.Box 218, Yorktown Heights, NY 10598, USA
Email: arconn@watson.ibm.com

[2] Rutherford Appleton Laboratory,
Computational Science and Engineering Departement
Chilton, Oxfordshire, England
Email: n.gould@rl.ac.uk
Current reports available from
http://www.numerical.rl.ac.uk/reports/reports.html

[3] C.E.R.F.A.C.S,
av. G. Coriolis, Toulouse, France
Email: Dominique.Orban@cerfacs.fr

[4] Department of Mathematics, University of Namur,
61, rue de Bruxelles, B-5000 Namur, Belgium,
Email: Philippe.Toint@fundp.ac.be
Current reports available from
ftp://thales.math.fundp.ac.be/pub/reports

A Primal-Dual Trust-Region Algorithm for Minimizing a Non-convex Function Subject to General Inequality and Linear Equality Constraints

Andrew R. Conn, Nicholas I. M. Gould, Dominique Orban and Philippe L. Toint

August 16, 1999

Abstract

A new primal-dual algorithm is proposed for the minimization of non-convex objective functions subject to general inequality and linear equality constraints. The method uses a primal-dual trust-region model to ensure descent on a suitable merit function. Convergence is proved to second-order critical points from arbitrary starting points. Preliminary numerical results are presented.

1 Introduction

In this paper, we consider algorithms for solving general (perhaps, non-convex), constrained, differentiable optimization problems. We shall distinguish between linear equality constraints and general inequality constraints. We thus consider the problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \\ \text{and} & c(x) \geq 0, \end{array} \tag{1.1}$$

where f is a real valued function of the variables $x \in \mathbb{R}^n$, A is an $m \times n$ matrix, b is a vector of $\mathbb{R}^m$, $c(x)$ a function from $\mathbb{R}^n$ into $\mathbb{R}^p$ and the inequalities are meant componentwise. An important instance of this problem is when $c(x) = x$, in which case the inequality constraints reduce to bound constraints.

At variance with our previous paper for the case $c(x) = x$, (Conn, Gould and Toint, 1999), we shall assume that we have a strictly feasible starting point x_0, i.e., that

AS.1 There is an x_0 such that $Ax_0 = b$ and $c(x_0) > 0$.

We do this for a number of reasons. In our experience, good primal-dual methods applied to the pure feasibility (phase-1) problem, when the only general nonlinear constraints are bound constraints, are usually very effective. Either a point satisfying AS.1 is rapidly determined in this case, or when this is not so, this is because the feasible region is small and thus the resulting point is close to its optimal value—of course, AS.1 may not hold, either because there is no feasible point, which will be detected by the phase-1 algorithm, or the feasible region has no relative interior, in which case it is sometimes possible to remove one or more offending constraints. More importantly, knowing a strictly feasible point leads to considerable simplifications over our previous algorithm. Indeed, most of the complications were due to the need to balance feasibility and objective improvement. Furthermore, by staying on the manifold $As = 0$, it is easier to ensure that the natural curvature of the problem is reflected in the direction-finding subproblems.

Besides covering general nonlinear inequality constraints instead of only bounds, this paper differs from its predecessor in another, significant way. The algorithm considered in our previous paper is of the linesearch variety. That is, a search direction is computed from the current estimate of the solution, and a suitable step then taken along this direction with the aim of reducing a merit function. The approach we consider here is an iterative trust-region method, in which the computation of search direction and step are combined. While in practice the two approaches often behave very similarly, a trust-region algorithm combines simplicity with strong convergence properties. In particular, trust-region methods can often be shown to be convergent to second-order critical points. It is these convergence guarantees that we find particularly attractive for non-convex problems.

Readers of our previous paper will also notice that we shall make a stronger distinction between the "outer" iteration, in which the parameters which define the particular merit function used are changed, and the "inner" iteration, in which a trust-region method is used to approximately minimize the merit function for a particular choice of the parameters. The distinction we use here makes it easier to distinguish the convergence of the inner iterates from the overall convergence of the method.

Not surprisingly, given the success of primal-dual interior point methods in linear programming, there has been considerable interest in extending such approaches to the general nonlinear case. We mention here some of the more recent work. Because of the increased complexity, details are important. In particular, the role of the merit function, treatment of indefinite Hessians and the implementation are critical. Yamashita, Yabe and Tanabe (1997), use a trust region method with exact second derivatives. Equality constraints are handled via an l_1 penalty and simple bounds by means of a log-barrier. Inequalities are converted to equalities with slack/surplus variables. They motivate taking a trust region approach by the need to handle indefinite Lagrangian Hessians. By contrast, Forsgren and Gill (1998) take a linesearch approach that uses a classical quadratic penalty and log barrier term to handle general equality and inequality constraints respectively, but augmented by terms that measure the proximity to the central path. Directions of negative curvature are determined via inertia controlling symmetric indefinite factorizations. Bakry, Tapia, Tsuchiya and Zhang (1996) use a line search framework and handle inequalities with slack/surplus variables. Their computational results are given with a merit function that is the l_2 norm of the residual for the first-order necessary conditions. Because of the tendency of this approach to converge to critical points that are not minima, Vanderbei and Shanno (1997) prefer using a merit function that handles the equality constraints as quadratic penalties and the slacks as barrier terms. Their context is also that of a line search, and indefiniteness is handled by modified Hessians. Gay, Overton and Wright (1998) also use a line search and handle indefiniteness using modified Hessians. Their merit function is a classical barrier function with an augmented Lagrangian to handle general equality constraints. In addition they use a watchdog technique. Finally, Byrd, Hribar and Nocedal (1997) use a sequential quadratic programming trust-region approach and a barrier function. Essentially, inequality constraints are transformed to equality constraints that are handled explicitly and the slacks are incorporated into the merit function as log-barrier terms. This problem is solved approximately (using multipliers corresponding to a shifted (augmented) Lagrangian plus the barrier function) with a merit function corresponding to a threshold on an l_∞ norm of the residual of the first order optimality conditions. This in turn is solved by means of an SQP method and the Byrd-Omojokun trust-region approach. Both primal and primal-dual versions are proposed.

2 Notation and assumptions

2.1 Basic notation and assumptions on the problem

Let $\mathcal{P} = \{x \mid c(x) \geq 0\}$ be the set of points satisfying the inequality constraints, $\mathcal{L} = \{x \mid Ax = b\}$ be the set of points satisfying the linear equality constraints, and so the intersection $\mathcal{F} \stackrel{\rm def}{=} \mathcal{P} \cap \mathcal{L}$ is the set of feasible points. Also let ri$\{\cdot\}$ denote the relative interior, which means that ri$\{\mathcal{P}\} = \{x \mid c(x) > 0\}$. If we denote the Euclidean inner product by $\langle \cdot, \cdot \rangle$ and let e be the vector of all ones, we shall assume that

$\boxed{\textbf{AS.2}}$ the functions $f(\cdot)$ and $c(\cdot)$ are twice continuously differentiable in their argument over some open set containing $\mathcal{F}$,

$\boxed{\textbf{AS.3}}$ the Hessian, $\nabla_{xx} f(x)$, of $f(x)$, as well as the Jacobian $J(x) \stackrel{\text{def}}{=} \nabla_x c(x)$ and each of the Hessians $\nabla_{xx} c_i(x)$ are uniformly bounded in Euclidean norm over $\mathcal{F}$,

$\boxed{\textbf{AS.4}}$ the matrix A has full rank, and

$\boxed{\textbf{AS.5}}$ the function $f(x) - \mu\langle e, \log(c(x))\rangle$ is bounded below on $\mathcal{F}$ for every $\mu > 0$.

Assumptions AS.2 and AS.3 simply ensure that $f(x)$ is well behaved in the region of interest. Since under AS.1, the constraints $Ax = b$ are consistent. AS.4 may be guaranteed by preprocessing the rows of A to remove redundancies (although we do not pretend that this is necessarily an easy task in practice). Assumption AS.5 might at first seem strong, but it is intended merely to rule out functions which grow more slowly at infinity than the log function. For such functions, the logarithmic barrier approach we consider in this paper is unlikely to succeed as the global minimizer of the barrier function is unbounded. In practice AS.5 can be expected to rule out few problems of interest.

In what follows, the i-th component of a vector x is denoted by $[x]_i$. We denote the diagonal matrix whose i-th diagonal is the i-th component of the vector $c(x)$, $c_i(x) = [c(x)]_i$, by $C(x)$. The n by n identity matrix is $\mathrm{diag}(e) = I$, and its i-th column is e_i. The vector g_k will be shorthand for $g(x_k)$, where $g(x)$ denotes the gradient of the objective function at x, $\nabla_x f(x)$. We let the columns of the n by $n - m$ matrix N be an orthonormal basis for the nullspace of A (so $AN = 0$ and $N^T N = I$). Finally, any continuous function $\omega : \mathbb{R}_+ \to \mathbb{R}_+$ is a said to be a *forcing* function if $\omega(\mu) = 0$ if and only if $\mu = 0$.

We denote the leftmost and rightmost eigenvalues of the symmetric matrix M by $\lambda^{\mathrm{L}}[M]$ and $\lambda^{\mathrm{R}}[M]$. Such a matrix is said to be *second-order sufficient* (with respect to A) if and only if the reduced matrix $N^T M N$ is positive definite (see, for instance, Gould, 1985). The matrix is *second-order necessary* if $N^T M N$ is positive semidefinite.

2.2 Norms

Because proper scaling is crucial in our algorithm, we need to consider a number of different norms whose purpose is to reflect the geometry of the problem. The first is simply the Euclidean ℓ_2 norm, which we shall denote by the symbol $\|\cdot\|$. For this norm, we have the relationship

$$\|X\| = \max_i \Big|[x]_i\Big| \leq \|x\|, \tag{2.1}$$

for any vector x. If S is a symmetric positive definite matrix, our second norm is the S norm of x, $\|x\|_S$, for which $\|x\|_S^2 = \langle x, Sx\rangle$.

It what follows, we shall choose to measure gradients and related quantities in a seminorm induced by a second-order sufficient iteration-dependent scaling matrix M_k, where k is the index of the current iteration of our algorithm. We define the *k-seminorm* of a vector g, $\|g\|_{[k]}$, by

$$\|g\|_{[k]}^2 \stackrel{\text{def}}{=} \langle y, g\rangle, \tag{2.2}$$

where y solves the system

$$\begin{pmatrix} M_k & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}.$$

This is actually a norm if g lies in the nullspace of A, and measures deviations from its range-space. In particular $\|g\|_{[k]} = 0$ if and only if $\|N^T g\| = 0$. It is easy to show that (2.2) may be expressed as

$$\|g\|_{[k]} = \|N^T g\|_{(N^T M_k N)^{-1}}. \tag{2.3}$$

In addition, because the gradients can be interpreted as linear forms on the space of the problem variables, it is natural to measure quantities directly involving these variables, such as the size of the trust region, in a seminorm corresponding to the dual of $\|\cdot\|_{[k]}$ in the nullspace of A. It is easy to verify that such a seminorm is given by $\|s\|_k \stackrel{\text{def}}{=} \|N^T s\|_{N^T M_k N}$ and is in fact a norm in the nullspace of A. As a consequence, for all $v, s \in \mathbb{R}^n$ such that $As = 0$, i.e. such that $s = NN^T s$, we have that

$$|\langle v, s\rangle| = |\langle (N^T M_k N)^{-\frac{1}{2}} N^T v, (N^T M_k N)^{\frac{1}{2}} N^T s\rangle| \le \|v\|_{[k]} \|s\|_k, \tag{2.4}$$

because of the Cauchy-Schwarz inequality. We stress that there is no need for M_k itself to be positive definite, merely that $N^T M_k N$ should be. If U is any symmetric matrix, we also define the reduced matrix

$$R[U, M_k] \stackrel{\text{def}}{=} (N^T M_k N)^{-\frac{1}{2}} N^T U N (N^T M_k N)^{-\frac{1}{2}},$$

its leftmost eigenvalue $\lambda^{\text{L}}_{M_k}[U] = \lambda^{\text{L}}[R[U, M_k]]$ and

$$\|U\|_{\{k\}} \stackrel{\text{def}}{=} \|R[U, M_k]\|.$$

We note that, again because of the Cauchy-Schwarz inequality,

$$|\langle s, Us\rangle| = |\langle (N^T M_k N)^{\frac{1}{2}} N^T s, R[U, M_k](N^T M_k N)^{\frac{1}{2}} N^T s\rangle| \le \|U\|_{\{k\}} \|s\|_k^2 \tag{2.5}$$

for every s such that $As = 0$. We also note that the inertia of $R[U, M_k]$ and $R[U, I] \equiv N^T U N$ are the same. In particular, we have that

$$\lambda^{\text{L}}_{M_k}[U] \ge 0 \text{ is equivalent to } \lambda^{\text{L}}_I[U] \ge 0. \tag{2.6}$$

We finally write $\|v\|_\diamond \stackrel{\text{def}}{=} \|N^T v\| = \|NN^T v\|$, the Euclidean norm of the projection of v onto the nullspace of A, and observe that $\|\cdot\|_\diamond$ is a self-dual norm in this nullspace.

3 The Algorithm

Our algorithm is basically a sequential minimization of a logarithmic barrier function subject to linear constraints, i.e. we propose to (approximately) solve

$$\begin{array}{ll} \text{minimize} & \phi(x, \mu_k) \\ \text{subject to} & Ax = b, \end{array} \tag{3.1}$$

where

$$\phi(x, \mu_k) = f(x) - \mu_k \langle e, \log((c(x))\rangle, \tag{3.2}$$

for a sequence of barrier parameters $\mu_k > 0$, $k = 1, 2, \ldots$, whose limiting value is zero. An approximate minimizer of problem (3.1), $\{x_{k+1}\}$, defines an *outer iterate*, and the associated adjustment of the barrier parameter and other tolerances defines the *outer iteration*. Outer iterations will be indexed by the subscript $k \ge 0$. Each outer iterate x_{k+1} is computed by using an appropriate *inner iteration* algorithm to approximately solve (3.1), with a corresponding sequence of *inner iterates* $\{x_{k,j}\}$. We now consider the inner and outer iterations in turn.

3.1 The inner iteration

We start by examining the inner iteration, whose purpose is to approximately solve (3.1) for a given value $\mu_k > 0$. The idea behind the algorithm we propose for this purpose is simply to apply a standard Newton-like trust-region method with the restriction that the iterates lie in the nullspace of A. At iteration (k, j), such a method would typically attempt to decrease the value of a quadratic model of the log-barrier function of the form

$$\begin{aligned} m_{k,j}(x_{k,j} + s) &= f(x_{k,j}) + \langle g_{k,j}, s\rangle + \tfrac{1}{2}\langle s, H_{k,j} s\rangle - \mu_k \langle e, \log(c(x_{k,j}))\rangle - \mu_k \langle J_{k,j}^T C_{k,j}^{-1} e, s\rangle \\ &\quad + \tfrac{1}{2}\mu_k \langle s, J_{k,j}^T C_{k,j}^{-2} J_{k,j} s\rangle - \tfrac{1}{2}\mu_k \sum_{i=1}^{p} \frac{1}{c_i(x_{k,j})} \langle s, \nabla_{xx} Q_{i,k,j} s\rangle, \end{aligned} \tag{3.3}$$

within a trust region, where the first three terms constitute a quadratic model of the objective function f with $H_{k,j}$ being an approximation of $\nabla_{xx} f(x_{k,j})$, where we write $J_{k,j} = J(x_{k,j})$, $C_{k,j} = C(x_{k,j})$ and where $Q_{i,k,j}$ approximates $\nabla_{xx} c_i(x_{k,j})$. However, when applying this method in practice, one often notices that convergence of the iterates $x_{k,j}$ slows down considerably whenever they happen to be close to the boundary of $\mathcal{P}$. This is because the singularity of the logarithm then plays a dominant role, which means that quadratic models of the log-barrier function, while very adequate locally, do not fit the barrier function well. One way of alleviating this numerical problem is to abandon the analytic expression for the local second-order behaviour of the barrier term and to replace it by a term whose growth would be, we hope, less dominant. In primal-dual methods, we choose to replace

$$H_{k,j} + \mu_k J_{k,j}^T C_{k,j}^{-2} J_{k,j} - \sum_{i=1}^{p} \frac{\mu_k}{c_i(x_{k,j})} \langle s, Q_{i,k,j} s\rangle \qquad \text{by} \qquad H_{k,j} + B_{k,j} - \sum_{i=1}^{p} [z_{k,j}]_i \langle s, Q_{i,k,j} s\rangle,$$

where

$$B_{k,j} \stackrel{\text{def}}{=} J_{k,j}^T C_{k,j}^{-1} Z_{k,j} J_{k,j} \tag{3.4}$$

for some bounded positive diagonal matrix $Z_{k,j}$. In other words, we consider the model

$$\begin{aligned} m_{k,j}(x_{k,j} + s) \quad = \quad & f(x_{k,j}) + \langle g_{k,j}, s\rangle + \tfrac{1}{2}\langle s, H_{k,j} s\rangle - \mu_k \langle e, \log(c(x_{k,j}))\rangle - \mu_k \langle J_{k,j}^T C_{k,j}^{-1} e, s\rangle \\ & + \tfrac{1}{2}\langle s, B_{k,j} s\rangle - \tfrac{1}{2} \sum_{i=1}^{p} [z_{k,j}]_i \langle s, \nabla_{xx} c_i(x_{k,j}) s\rangle \end{aligned}$$

instead of (3.3). Defining

$$G_{k,j} \stackrel{\text{def}}{=} H_{k,j} - \sum_{i=1}^{p} [z_{k,j}]_i Q_{i,k,j}, \tag{3.5}$$

we obtain that our model has the form

$$m_{k,j}(x_{k,j} + s) = \phi(x_{k,j}, \mu_k) + \langle g_{k,j} - \mu_k J_{k,j}^T C_{k,j}^{-1}, s\rangle + \tfrac{1}{2}\langle s, [G_{k,j} + B_{k,j}] s\rangle. \tag{3.6}$$

Note that $G_{k,j}$ is an approximation of the Hessian of the Lagrangian function

$$\psi(x, z) = f(x) - \langle z, c(x)\rangle$$

at $(x_{k,j}, z_{k,j})$ with respect to x, that is $G_{k,j} \approx \nabla_{xx} \psi(x_{k,j}, z_{k,j})$.

Interestingly, as is well-known, there is another way to motivate this modification of the barrier model's Hessian, using a perturbation argument. Consider the first-order necessary conditions for the problem of minimizing the model of the objective function on the feasible set, namely

$$g(x) + A^T y - J(x)^T z = 0, \quad Ax = b, \quad C(x) z = 0, \quad c(x) \geq 0, \quad z \geq 0, \tag{3.7}$$

where z is the vector of dual variables (Lagrange multipliers) for the inequality constraints and y is the vector of Lagrange multipliers associated with the equality constraints. The third equation of (3.7) is known as the problem's *complementarity condition.* Notice that it expresses a true combinatorial requirement: "if a constraint is non-zero, then its corresponding dual variable must be zero" and vice-versa. As combinatorial conditions may be very hard to verify, especially for large problems, we perturb them. Introducing a small perturbation parameter $\mu > 0$, we then write

$$g(x) + A^T y - J(x)^T z = 0, \quad Ax = b, \quad C(x) z = \mu e, \quad c(x) \geq 0, \quad z \geq 0.$$

Newton's equation for this system of nonlinear equations at some iterate $(x_{k,j}, z_{k,j})$ and for some value μ_k of the perturbation parameter are

$$\begin{aligned} G_{k,j} \Delta x_{k,j} + A^T y_{k,j+1} - J_{k,j}^T \Delta z_{k,j} &= -g_{k,j} + z_{k,j}, \\ A \Delta x_{k,j} &= 0 \\ C_{k,j} \Delta z_{k,j} + Z_{k,j} J_{k,j} \Delta x_{k,j} &= \mu_k e - C_{k,j} Z_{k,j} e, \end{aligned} \tag{3.8}$$

where $Z_{k,j} = \operatorname{diag}([z_{k,j}]_1, \dots, [z_{k,j}]_n)$ and where we have written $y_{k,j+1} = y_{k,j} + \Delta y_{k,j}$. Ignoring the non-negativity conditions and eliminating $\Delta z_{k,j}$ in (3.8), we obtain the system

$$\begin{pmatrix} G_{k,j} + B_{k,j} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x_{k,j} \\ y_{k,j+1} \end{pmatrix} = - \begin{pmatrix} g_{k,j} - \mu_k J_{k,j}^T C_{k,j}^{-1} e \\ 0 \end{pmatrix} \tag{3.9}$$

and

$$\Delta z_{k,j} = -z_{k,j} - C_{k,j}^{-1} Z_{k,j} J_{k,j} \Delta x_{k,j} + \mu_k C_{k,j}^{-1} e. \tag{3.10}$$

We then note that the first component of right-hand side of this relation is nothing but the negative gradient of the log-barrier function, $-\nabla_x \phi(x, \mu_k)$. Moreover, these equations are precisely the first-order optimality conditions for the problem of minimizing the model (3.6), subject to the constraints $A\Delta x_{k,j} = 0$. Hence $\Delta x_{k,j}$ may be interpreted as a constrained Newton-type step for $\phi(x, \mu_k)$. This is exactly what we proposed above, and we would like to emphasize that we now interpret $z_{k,j}$ as the vector of dual variables.

We may therefore wish to compute the step from (3.9)-(3.10), but some additional precautions are necessary. Note that (3.9) fully defines $\Delta x_{k,j}$, and $y_{k,j+1}$ provided AS.4 holds and the matrix $G_{k,j} + B_{k,j}$ is nonsingular on the nullspace of A. This is obviously the case if $f(x)$ is strictly convex, but may not be true in general. More significantly, (3.9) is inappropriate if $G_{k,j} + B_{k,j}$ is not second-order sufficient, as then $\Delta x_{k,j}$ at best defines a saddle point for the model. Thus the model should either be modified, as we proposed for the case of bound constraints in a previous paper (Conn et al., 1999), or restricted by a trust-region constraint, as we propose here. Observe also that, if $\Delta x_{k,j}$ is well defined, $\Delta z_{k,j}$ is in turn well defined by (3.10). Of course, there is no automatic guarantee that $(c(x_{k,j} + \Delta x_{k,j}), z_{k,j} + \Delta z_{k,j}) > 0$ so we would need to be careful before allowing such a step. Moreover, the fact that $b(x, \mu_k) = \mu_k \langle e, \log(c(x)) \rangle$ is undefined wherever x does not belong to $\operatorname{ri}\{\mathcal{P}\}$ creates a difficulty, for nothing in the above derivation prevents from predicting a step $\Delta x_{k,j}$ such that $x_{k,j} + \Delta x_{k,j} \notin \operatorname{ri}\{\mathcal{P}\}$. The value $b(x_{k,j} + \Delta x_{k,j}, \mu_k)$, and therefore $\phi(x_{k,j} + \Delta x_{k,j}, \mu_k)$, are then undefined, and the algorithm breaks down. Fortunately, such undesirable algorithmic behaviour can be circumvented quite simply. The idea is to observe that, if $x_{k,j} + \Delta x_{k,j}$ lies outside $\mathcal{P}$, this is merely an indication that the model $m_{k,j}$ does not approximate the objective $\phi(x_{k,j} + s, \mu_k)$ very well. In particular, this indicates that a smaller step from $x_{k,j}$ (which must lie inside $\mathcal{P}$) is necessary. A simple technique is to restrict the trust-region radius enough to ensure that $x_{k,j} + \Delta x_{k,j} \in \operatorname{ri}\{\mathcal{P}\}$, which must occur when $\Delta_{k,j}$ is small enough to enforce that

$$\mathcal{B}_{k,j} \stackrel{\text{def}}{=} \{x_{k,j} + s \in \mathbb{R}^n \mid As = 0 \ \text{ and } \ \|s\|_{k,j} \le \Delta_{k,j}\} \subset \operatorname{ri}\{\mathcal{P}\}.$$

The crucial point is that this restriction may be decided without even trying to compute the (undefined) function value at $x_{k,j} + s_{k,j}$, therefore avoiding the situation where the algorithm breaks down. Thus iteration j is viewed as unsuccessful and $\Delta_{k,j}$ is reduced whenever $x_{k,j} + \Delta x_{k,j}$ falls in the region where the barrier function is undefined. If this is not the case, the trial step $s_{k,j} = \Delta x_{k,j}$ is acceptable.

It is important to notice that we are prepared to solve the trust-region subproblem

$$\begin{array}{ll} \text{minimize} & m_{k,j}(x_{k,j} + s) \\ \text{subject to} & As = 0 \\ \text{and} & \|s\|_{k,j} \le \Delta_{k,j}, \end{array} \tag{3.11}$$

only approximately, in that we merely aim to improve $m_{k,j}(x_{k,j} + s)$ while satisfying the remaining constraints. In particular, there is no evidence in general that finding an accurate solution is especially beneficial. Thus, we may be satisfied to find an approximation which guarantees convergence, knowing that any extra effort may be expended when necessary. To this end, we assume that the step $s_{k,j}$ is chosen so that

$$\begin{aligned} m_{k,j}(x_{k,j} + s_{k,j}) &\le m_{k,j}(x_{k,j}) \\ &-\theta \max\left\{ \|\nabla_x \phi(x_{k,j}, \mu_k)\|_{[k,j]} \min\left[\frac{\|\nabla_x \phi(x_{k,j}, \mu_k)\|_{[k,j]}}{\beta_{k,j}}, \Delta_{k,j}, \right], -\tau_{k,j} \min\left[\tau_{k,j}^2, \Delta_{k,j}^2\right] \right\} \end{aligned} \tag{3.12}$$

where $\theta \in (0, \frac{1}{2})$,

$$\beta_{k,j} = 1 + \|G_{k,j} + B_{k,j}\|_{\{k,j\}} \quad \text{and} \quad \tau_{k,j} = \lambda^{\text{L}}_{M_{k,j}}[G_{k,j} + B_{k,j}]. \tag{3.13}$$

This assumption is usual in trust-region methods. Because $\beta_{k,j}$ gives a bound on the curvature of the quadratic model, in the reduced space and scaled, the first term in the maximum guarantees that the model reduction is at least a fraction of that obtained at the Cauchy point, while the second term ensures that negative curvature is exploited when present. Projected conjugate-gradient/Lanczos-like methods are able to produce such a step at a reasonable cost (see Gould, Lucidi, Roma and Toint, 1999).

The actual choice of norm in the second constraint of (3.11) is important. We believe that the norm defining the trust-region shape should reflect the underlying geometry of the problem, and the freedom of choice of the matrix $M_{k,j}$ defining this norm will allow us to capture this geometry. A natural choice in this context is to choose $M_{k,j} = \nabla_{xx} m_{k,j}(x_{k,j}) = G_{k,j} + B_{k,j}$. However, this matrix may not be second-order sufficient, in which case we may have to modify $G_{k,j}$ to ensure this property (remember that, by definition, $B_{k,j}$ is positive semidefinite). To reflect this possible modification, we define

$$M_{k,j} = W_{k,j} + B_{k,j}, \tag{3.14}$$

where, for instance, $W_{k,j} = G_{k,j}$ whenever $G_{k,j} + B_{k,j}$ is second-order sufficient.

The algorithm that we propose for the inner iterations is presented as Algorithm 3.1.

Algorithm 3.1: Inner iteration

Step 0: Initialization. An initial point $x_{k,0} \in \text{ri}\{\mathcal{P}\} \cap \mathcal{L}$, a vector $z_{k,0} > 0$ of dual variables and an initial trust-region radius $\Delta_{k,0}$ are given. The constants $\varsigma_k, \eta_1, \eta_2, \gamma_1$, and γ_2 are also given and satisfy the conditions $0 < \varsigma_k < 1$, $0 < \eta_1 \le \eta_2 < 1$ and $0 < \gamma_1 \le \gamma_2 < 1$. Compute $f(x_{k,0})$ and $c(x_{k,0})$ (if not already known) and set $j = 0$.

Step 1: Model definition. Choose the scaling matrix $M_{k,j}$ according to (3.14) and define, in $\mathcal{B}_{k,j}$, a model $m_{k,j}$ of $\phi(x_{k,j} + s, \mu_k)$ which is of the form (3.6).

Step 2: Step calculation. Compute a step $s_{k,j}$ such that $x_{k,j} + s_{k,j} \in \mathcal{B}_{k,j}$ and such that it sufficiently reduces the model $m_{k,j}$ in the sense of (3.12).

Step 3: Acceptance of the trial point. If

$$c(x_{k,j} + s_{k,j}) \ge \varsigma_k c(x_{k,j}), \tag{3.15}$$

compute $\phi(x_{k,j} + s_{k,j}, \mu_k)$ and define the ratio

$$\rho_{k,j} = \frac{\phi(x_{k,j}, \mu_k) - \phi(x_{k,j} + s_{k,j}, \mu_k)}{m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j} + s_{k,j})};$$

else set $\rho_{k,j} = -\infty$. Then if $\rho_{k,j} \ge \eta_1$, define $x_{k,j+1} = x_{k,j} + s_{k,j}$; otherwise define $x_{k,j+1} = x_{k,j}$.

Step 4: Trust-region radius update. Set

$$\Delta_{k,j+1} \in \begin{cases} [\Delta_{k,j}, \infty) & \text{if } \rho_{k,j} \ge \eta_2, \\ [\gamma_2 \Delta_{k,j}, \Delta_{k,j}] & \text{if } \rho_{k,j} \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_{k,j}, \gamma_2 \Delta_{k,j}] & \text{if } \rho_{k,j} < \eta_1. \end{cases}$$

Step 5: Update the dual variables. Define $z_{k,j+1} > 0$. Increment j by one and go to Step 1.

The only differences between this algorithm and a standard trust-region method, besides the fact that the objective function is now $\phi(x, \mu_k)$ instead of $f(x)$ and we are accounting for the linear equality constraints by working in the corresponding reduced space, are the requirement that the initial point must lie in $\mathcal{L}$ and the interior of $\mathcal{P}$ and the fact that an iterate is rejected if (3.15) does not hold. We have intentionally not specified how the parameter ς_k is chosen for each inner minimization. This parameter specifies the minimum relative value of the inequality constraints which is acceptable in the course of the current minimization. The fact that is not fixed but may itself tend to zero as k increases makes fast asymptotic convergence of the outer iterates possible, but we do not discuss this question in detail. Also note that the possibility of choosing $\Delta_{k,j+1}$ as large as one wishes on successful iterations may be important in practice, because it allows the trust-region radius to return to a reasonable value as soon as a successful step is made, instead of being constrained to remain of the order of magnitude of the distance of $x_{k,j}$ to the boundary of $\mathcal{P}$.

Iterations at which $\rho_{k,j} \geq \eta_1$, and thus the current iterate is redefined, are called successful. We denote by $\mathcal{S}$ the set consisting of the indices of all successful iterations.

3.2 The outer iteration

After describing the mechanism of the inner iterations for finding an approximate minimizer of (3.1), we now consider the outer iteration to solve (1.1), which we formally state as Algorithm 3.2.

Algorithm 3.2: Outer iteration

Step 0: Initialization. An initial point $x_0 > 0$ that satisfies $Ax_k = b$, a vector of initial dual variables $z_0 > 0$ and an initial barrier parameter $\mu_0 > 0$ are given. The forcing functions $\epsilon^{\mathrm{C}}(\mu)$, $\epsilon^{\mathrm{D}}(\mu)$ and $\epsilon^{\mathrm{E}}(\mu)$ are also given. Set $k = 0$.

Step 1: Inner minimization Choose a value $\varsigma_k \in (0,1)$. Minimize the log-barrier function $\phi(x,\mu_k) = f(x) - \mu_k\langle e, \log(c(x))\rangle$ starting from x_k. Stop this inner algorithm as soon as an iterate $(x_{k,j}, z_{k,j}) = (x_{k+1}, z_{k+1})$ is found such that

$$
\begin{aligned}
Ax_{k+1} &= b && (3.16)\\
(c(x_{k+1}), z_{k+1}) &> 0 && (3.17)\\
\|C(x_{k+1})Z_{k+1} - \mu_k I\| &\leq \epsilon^{\mathrm{C}}(\mu_k) && (3.18)\\
\|g_{k+1} - J_{k+1}^T z_{k+1}\|_{[k+1]} &\leq \epsilon^{\mathrm{D}}(\mu_k) \quad \text{and} && (3.19)\\
\lambda^{\mathrm{L}}_{M_{k+1}}\left[G_{k+1} + B_{k+1}\right] &\geq -\epsilon^{\mathrm{E}}(\mu_k), && (3.20)
\end{aligned}
$$

where $M_{k+1} = M_{k,j}$. Increment k by one, and repeat Step 1.

Our intention is to find a point which satisfies (3.16)–(3.20) by applying Algorithm 3.1 to approximately solve (3.1), assuming for now that it converges to a second-order critical point, that is a point at which first- and second-order necessary optimality hold, for this subproblem. For, if $x_{k,*}$ were such a point, the conditions

$$Ax_{k,*} = b \tag{3.21}$$

$$c(x_{k,*}) > 0 \tag{3.22}$$

$$N^T\nabla_x\phi(x_{k,*},\mu_k) \equiv N^T\left(g(x_{k,*}) - \mu_k J(x_{k,*})^T C(x_{k,*})^{-1}e\right) = 0 \quad \text{and} \tag{3.23}$$

$$\lambda^{\mathrm{L}}\left[N^T\nabla_{xx}\phi(x_{k,*},\mu_k)N\right] \geq 0 \tag{3.24}$$

must occur. On defining

$$z_{k,*} = \mu_k C(x_{k,*})^{-1}e > 0, \tag{3.25}$$

we see from (3.23) that

$$N^T\left(g(x_{k,*}) - J(x_{k,*})^T z_{k,*}\right) = 0,$$

and by definition

$$C(x_{k,*})Z_{k,*} - \mu_k I = 0.$$

Moreover

$$\nabla_{xx}\phi(x_{k,*},\mu_k) = \nabla_{xx} f(x_{k,*}) + \mu_k J(x_{k,*})^T C(x_{k,*})^{-2} J(x_{k,*}) - \mu_k \sum_{i=1}^{p} \frac{1}{c_i(x_{k,*})} \nabla_{xx} c_i(x_{k,*})$$

and thus, taking (3.25) into account and assuming that G_k converges to $\nabla_{xx}\psi(x_{k,*}, z_{k,*})$, we see that $G_k + B_k$ converges to $\nabla_{xx}\phi(x_{k,*},\mu_k)$. Combining these conclusions, we therefore obtain that any inner iterate sufficiently close to $x_{k,*}$ provides a suitable terminating value satisfying (3.16)–(3.20).

We should also add a comment on the terminating condition (3.20). The aim here is to ensure that second-order necessary conditions for the solution of (1.1) are implied by requiring that similar conditions hold for (3.1). However, one naturally expects that second-order conditions for (3.1) would involve the matrix

$$N^T \nabla_{xx}\phi(x_k,\mu_k) N \tag{3.26}$$

not

$$N^T\left(G_k + B_k\right) N. \tag{3.27}$$

The reason we base our terminating condition (3.20) on (3.27) rather than (3.26) is simply that Algorithm 3.1 uses this matrix rather than (3.26) at its core — spectral information will thus be conveniently available for (3.27) but not for (3.26). Of course (3.26) and (3.27) coincide when z_k is defined via (3.25), and the two matrices can be expected to be close when $\epsilon^C(\mu_k)$ in (3.18) is small.

The variables z_k computed by the algorithm are estimates of the dual variables associated with the inequality constraints at a solution of (1.1). The particular choice (3.25) is appropriate at a critical point of (3.1), while it is less suitable away from such a critical point. As we shall see, there are better choices in this latter case.

The seminorm used in (3.19) is the appropriate measure of convergence of the gradient since, as we mentioned above, the trust region is defined in the dual to this seminorm, $\|\cdot\|_{k+1}$ in the nullspace of A. At first sight, we may question whether it is reasonable to expect global convergence properties of both the inner and outer algorithms if we use the scaled norms. The question arises because the matrices M_{k+1} blow up when, as is highly likely, the iterates approach the boundary of the feasible set. It is fortunate that global convergence to critical points may still be proved with the scaled formulations, as we will shortly see.

4 Convergence theory

In this section, we consider the convergence of Algorithm 3.2, where we intend to use the inner-iteration Algorithm 3.1 to calculate each of the iterates.

4.1 Further assumptions

In order to prove the desired results, we must state our assumptions on the dual variables and on the matrix $G_{k,j}$.

AS.6 For each $k \ge 0$, there exists a constant $\kappa_{\rm zi}(k) > 0$ such that, for all $j \ge 0$ and all $i = 1,\ldots,p$,

$$[z_{k,j}]_i \le \kappa_{\rm zi}(k) \max\left[\frac{1}{c_i(x_{k,j})}, 1\right].$$

AS.7 the approximate Hessian of the Lagrangian remains bounded, i.e.

$$\|G_{k,j}\|_{\{k,j\}} \leq \kappa_G$$

for all $k, j \geq 0$, and for some $\kappa_G > 0$,

Note that , because of AS.2, AS.3 and AS.6, AS.7 is automatically satisfied if the appropriate exact values are chosen for $H_{k,j}$ and $Q_{i,k,j}$. We finally state the assumptions on the scaling matrices and require that

AS.8 there exists $\epsilon_M \in (0,1)$ and $\kappa_{\rm W} > 0$ such that, for all k and all j, the scaling matrix $M_{k,j} = W_{k,j} + B_{k,j}$ and its component $W_{k,j}$ satisfy

$$\lambda_{\min}[N^T M_{k,j} N] \geq \epsilon_M \tag{4.1}$$

and

$$\|N^T W_{k,j} N\| \leq \kappa_{\rm W}. \tag{4.2}$$

As a consequence of the first part of this last assumption, we note that

$$\|U\|_{\{k,j\}} = \|(N^T M_{k,j} N)^{-\frac{1}{2}} N^T U N (N^T M_{k,j} N)^{-\frac{1}{2}}\| \leq \frac{1}{\epsilon_M}\|U\| \tag{4.3}$$

for every symmetric matrix U.

4.2 Convergence of the inner iteration

We first prove that conditions (3.16)–(3.20) will eventually be satisfied after a finite number of iterations of Algorithm 3.1. The main idea is that we may apply a variation on a traditional trust-region algorithm for unconstrained optimization in the subspace $\mathcal{L}$ of all vectors satisfying the linear constraints. Unless otherwise stated, we assume in this section that $\epsilon^{\rm C} = \epsilon^{\rm D} = \epsilon^{\rm E} = 0$.

We start our analysis by showing that, as expected, the iterates generated by Algorithm 3.1 will never become infinitely close to the boundary of $\mathcal{P}$.

Lemma 4.1 Suppose that AS.1–AS.5 hold, and that $\{x_{k,j}\}$ is a sequence of iterates generated by Algorithm 3.1. Then there exists a constant $\kappa_b(k) \in (0,1)$ depending only on k such that, for all j,

$$\min_{i=1,\ldots,n} c_i(x_{k,j}) \geq \kappa_b(k), \quad (i = 1, \ldots, p).$$

Proof. Clearly, the level set $\{x \in \mathcal{P} \mid b(x,\mu) \leq b(x_{k,0},\mu)\}$, and thus of $\phi(x_{k,0},\mu)$, must be bounded away from $\partial\mathcal{P}$. The existence of $\kappa_b(k)$ then results from the inequality $\phi(x_{k,j},\mu_k) \leq \phi(x_{k,0},\mu_k)$ which is true for all $j \geq 0$. Moreover, it can always be chosen small enough to ensure that it belongs to $(0,1)$. □

This result is crucial because it states that all arguments that use a sequence of trust-region radii $\Delta_{k,j}$ converging to zero will not be hindered by the restriction of remaining in the interior of $\mathcal{P}$. Note that AS.6 and Lemma 4.1 together ensure that, for fixed k and all i and j,

$$[z_{k,j}]_i \leq \frac{\kappa_{\rm zi}(k)}{\kappa_b(k)} \stackrel{\rm def}{=} \kappa_z(k), \tag{4.4}$$

where $\kappa_z(k)$ only depends on k. Also note that the first part of (3.13), the triangle inequality, (4.3), AS.7, Lemma 4.1 and (4.4) together imply that, for all k and j,

$$\beta_{k,j} \leq 1 + \|G_{k,j}\|_{\{k,j\}} + \|B_{k,j}\|_{\{k,j\}} \leq 1 + \kappa_G + \frac{\kappa_z(k)\kappa_J^2}{\epsilon_M \kappa_b(k)} \stackrel{\rm def}{=} \kappa_\beta(k), \tag{4.5}$$

where $\kappa_J > 0$ is the upper bound on $\|J(x)\|$ implied by AS.3. The bound (4.4) is important because it guarantees, together with Lemma 4.1, that all scaled norms used during a single inner minimization are uniformly equivalent, as we now show.

Lemma 4.2 Suppose that $\{x_{k,j}\}$ is a sequence of iterates generated by Algorithm 3.1 and that AS.1–AS.6 hold. Suppose furthermore that $M_{k,j}$ satisfies AS.8. Then there exists a constant $\kappa_{\rm n}(k) \geq 1$ only depending on k such that, for all j (and fixed k) the seminorms $\|\cdot\|_{k,j}$ and $\|\cdot\|_{[k,j]}$ satisfy

$$\frac{1}{\kappa_{\rm n}(k)}\|v\|_{k,j} \leq \|v\|_\diamond \leq \kappa_{\rm n}(k)\|v\|_{k,j},$$

and

$$\frac{1}{\kappa_{\rm n}(k)}\|v\|_{[k,j]} \leq \|v\|_\diamond \leq \kappa_{\rm n}(k)\|v\|_{[k,j]},$$

for all $v \in \mathbb{R}^n$.

Proof. We start by proving the first series of inequalities. First notice that the result obviously holds if $N^Tv = 0$. We therefore restrict our attention to vectors $N^Tv \neq 0$. Suppose first that

$$\langle N^Tv, (N^TW_{k,j}N)N^Tv\rangle \leq \langle N^Tv, (N^TB_{k,j}N)N^Tv\rangle. \tag{4.6}$$

Then, using (4.4), Lemma 4.1 and AS.3,

$$\begin{aligned} \|v\|_{k,j}^2 = \|N^Tv\|^2_{N^TM_{k,j}N} &= \langle N^Tv, N^T[W_{k,j}+B_{k,j}]NN^Tv\rangle \\ &\leq 2\langle NN^Tv, (J_{k,j}^TC_{k,j}^{-1}Z_{k,j}J_{k,j})NN^Tv\rangle \\ &\leq \frac{2\kappa_z(k)\kappa_J^2}{\kappa_b(k)}\|NN^Tv\|^2 \\ &= \frac{2\kappa_z(k)\kappa_J^2}{\kappa_b(k)}\|v\|_\diamond^2. \end{aligned} \tag{4.7}$$

If, on the other hand, (4.6) does not hold, then

$$\begin{aligned} \|v\|_{k,j}^2 &= \langle N^Tv, N^T[W_{k,j}+B_{k,j}]N^Tv\rangle \\ &\leq 2\langle N^Tv, (N^TW_{k,j}N)N^Tv\rangle \\ &\leq 2\kappa_{\rm W}\|N^Tv\|^2 \\ &= 2\kappa_{\rm W}\|v\|_\diamond^2, \end{aligned} \tag{4.8}$$

because of (4.2). Combining (4.7) and (4.8), we obtain that

$$\min\left[\frac{\kappa_b(k)}{2\kappa_z(k)\kappa_J^2}, \frac{1}{2\kappa_{\rm W}}\right]\|v\|_{k,j}^2 \leq \|v\|_\diamond^2. \tag{4.9}$$

Turning to the other inequality for the seminorm $\|\cdot\|_{k,j}$, (4.1) implies that, for all $v \neq 0$,

$$\begin{aligned} \frac{\|v\|_\diamond^2}{\|v\|_{k,j}^2} &= \frac{\langle (N^TM_{k,j}N)^{-\frac{1}{2}}[(N^TM_{k,j}N)^{\frac{1}{2}}N^Tv], (N^TM_{k,j}N)^{-\frac{1}{2}}[(N^TM_{k,j}N)^{\frac{1}{2}}N^Tv]\rangle}{\langle (N^TM_{k,j}N)^{\frac{1}{2}}N^Tv, (N^TM_{k,j}N)^{\frac{1}{2}}N^Tv\rangle} \\ &\leq \|(N^TM_{k,j}N)^{-1}\| \\ &\leq \tfrac{1}{\epsilon_M}. \end{aligned}$$

This inequality and (4.9) together prove the desired inequality for the $\|\cdot\|_{k,j}$ seminorm with

$$\kappa_{\rm n}(k) \stackrel{\rm def}{=} \sqrt{\max\left[\frac{1}{\epsilon_M}, \frac{2\kappa_z(k)\kappa_J^2}{\kappa_b(k)}, 2\kappa_{\rm W}\right]}.$$

The proof of the second set of inequalities in the theorem is obtained by a similar argument involving $(N^T M_{k,j} N)^{-1}$ instead of $N^T M_{k,j} N$, since the eigenvalues of the former are then contained in the interval

$$\left[\min\left[\frac{\kappa_b(k)}{2\kappa_z(k)\kappa_J^2}, \frac{1}{2\kappa_{\rm W}}\right], \frac{1}{\epsilon_M}\right] \quad \text{instead of} \quad \left[\epsilon_M, \max\left[\frac{2\kappa_z(k)\kappa_J^2}{\kappa_b(k)}, 2\kappa_{\rm W}\right]\right]$$

for the latter. □

A last useful consequence of Lemma 4.1 is that there is a neighbourhood of each iterate $x_{k,j}$ whose diameter only depends on k such that (3.15) holds in this neighbourhood.

Lemma 4.3 Suppose that AS.1–AS.6 and AS.8 hold, and that $\{x_{k,j}\}$ is a sequence of iterates generated by Algorithm 3.1. Then there exists a constant $\kappa_x(k) \in (0,1)$ depending only on k such that, for all j,

$$c_i(w) \geq \varsigma_k c_i(x_{k,j}) \quad (i = 1, \ldots, p)$$

for every $w \in \mathcal{F}$ such that

$$\|w - x_{k,j}\|_{k,j} \leq \kappa_x(k).$$

Proof. Assume, for the purpose of obtaining a contradiction that the exists some $w \in \mathcal{F}$, some $i \in \{1, \ldots, p\}$ and some iterate $x_{k,j}$ generated by Algorithm 3.1 such that

$$\|w - x_{k,j}\|_{k,j} \leq \frac{(1-\varsigma_k)\kappa_b(k)}{2\kappa_{\rm n}(k)\kappa_J} \stackrel{\rm def}{=} \kappa_x(k) \tag{4.10}$$

and

$$c_i(w) < \varsigma_k c_i(x_{k,j}) \tag{4.11}$$

for some $i \in \{1, \ldots, p\}$. Let $v \in [x_{k,j}, w]$ be the point in that segment which is such that $c_i(v) = \varsigma_k c_i(x_{k,j})$ and which is closest (in the $\|\cdot\|_{k,j}$ seminorm) to $x_{k,j}$. Note that v must exist because of AS.2 and is unique because of AS.8. AS.2 also implies that

$$\varsigma_k c_i(x_{k,j}) = c_i(v) = c_i(x_{k,j}) + \langle \nabla_x c_i(\xi), v - x_{k,j}\rangle \geq c_i(x_{k,j}) - \|\nabla_x c_i(\xi)\|_{[k,j]}\|v - x_{k,j}\|_{k,j} \tag{4.12}$$

for some $\xi \in [x_{k,j}, v]$, where we used (2.4) to deduce the last inequality. But the definition of v and the inclusion $x_{k,j} \in \mathcal{F}$ imply that the segment $[x_{k,j}, v]$ is also included in $\mathcal{F}$. Hence $\xi \in \mathcal{F}$ and we may apply AS.3 and Lemma 4.2 to bound $\|\nabla_x c_i(\xi)\|_{[k,j]}$ above by $\kappa_{\rm n}(k)\kappa_J$, which implies, using (4.12), the bound $\|v - x_{k,j}\|_{k,j} \leq \|w - x_{k,j}\|_{k,j}$, Lemma 4.1 and (4.10), that

$$0 \geq (1-\varsigma_k)c_i(x_{k,j}) - \kappa_{\rm n}(k)\kappa_J\|v - x_{k,j}\|_{k,j} \geq (1-\varsigma_k)\kappa_b(k) - \kappa_{\rm n}(k)\kappa_J\|w - x_{k,j}\|_{k,j} \geq \tfrac{1}{2}(1-\varsigma_k)\kappa_b(k) > 0,$$

which is impossible. Hence no such w, i and j can exist and the lemma is proved. □

We now consider the error between the predicted and the exact objective value at the trial point as follows.

Theorem 4.4 Assume that AS.1–AS.8 hold. Assume also that $s_{k,j}$ is generated as in Algorithm 3.1 and that

$$\Delta_{k,j} \leq \kappa_x(k) \tag{4.13}$$

then we have that,

$$|\phi(x_{k,j}+s_{k,j},\mu_k) - m_{k,j}(x_{k,j}+s_{k,j})| \leq \kappa_\phi(k)\Delta_{k,j}^2. \tag{4.14}$$

where

$$\kappa_\phi(k) \stackrel{\rm def}{=} \tfrac{1}{2}\kappa_G + \frac{1}{2\epsilon_M}\left[\kappa_f + \frac{\mu_k\kappa_J^2}{\varsigma_k^2\kappa_b(k)^2} + p\frac{\mu_k\kappa_c}{\kappa_b(k)} + \frac{\kappa_z(k)\kappa_J^2}{\kappa_b(k)}\right], \tag{4.15}$$

where κ_f and κ_c are, respectively, the upper bounds on $\|\nabla_{xx}f(x)\|$ and $\|\nabla_{xx}c_i(x)\|$ implied by AS.3.

Proof. Taking the difference of the second-order Taylor's expansion of ϕ and $m_{k,j}$ and considering absolute values yields that, for some $\xi_{k,j}$ in $[x_{k,j}, x_{k,j}+s_{k,j}]$,

$$|\phi(x_k+s_k,\mu_k) - m_{k,j}(x_{k,j}+s_{k,j})| = \tfrac{1}{2}|\langle s_{k,j}, \nabla_{xx}\phi(\xi_{k,j},\mu_k)s_{k,j}\rangle - \langle s_{k,j}, \nabla_{xx}m_{k,j}(x_{k,j})s_{k,j}\rangle|, \tag{4.16}$$

because of AS.2 and (3.6). Lemma 4.3, the bound $\|s_{k,j}\|_{k,j} \leq \Delta_{k,j}$ and (4.13) then ensure that (3.15) holds and that the segment $[x_{k,j}, x_{k,j}+s_{k,j}]$ belongs to $\mathcal{F}$. Thus AS.1, AS.3 and (4.3) imply, using Lemma 4.1, that

$$\begin{aligned}
\|\nabla_{xx}\phi(\xi_{k,j},\mu_k)\|_{\{k,j\}} &\leq \|\nabla_{xx}f(\xi_{k,j})\|_{\{k,j\}} + \mu_k\|J(\xi_{k,j})^T C(\xi_{k,j})^{-2}J(\xi_{k,j})\|_{\{k,j\}} \\
&\qquad + \mu_k\sum_{i=1}^p \frac{1}{c_i(x_{k,j})}\|\nabla_{xx}c_i(\xi_{k,j})\|_{\{k,j\}} \\
&\leq \frac{1}{\epsilon_M}\Big[\|\nabla_{xx}f(\xi_{k,j})\| + \mu_k\|J(\xi_{k,j})^T C(\xi_{k,j})^{-2}J(\xi_{k,j})\| \\
&\qquad + \mu_k\sum_{i=1}^p \frac{1}{c_i(x_{k,j})}\|\nabla_{xx}c_i(\xi_{k,j})\|\Big] \\
&\leq \frac{1}{\epsilon_M}\left[\kappa_f + \frac{\mu_k\kappa_J^2}{\varsigma_k^2\kappa_b(k)^2} + p\frac{\mu_k\kappa_c}{\varsigma_k\kappa_b(k)}\right] \\
&\stackrel{\rm def}{=} \kappa_1(k).
\end{aligned}$$

Similarly, using (4.4) and (3.6), we obtain that

$$\|\nabla_{xx}m_{k,j}(x_{k,j})\|_{\{k,j\}} \leq \|G_{k,j}\|_{\{k,j\}} + \|B_{k,j}\|_{\{k,j\}} \leq \kappa_G + \frac{\kappa_z(k)\kappa_J^2}{\epsilon_M\kappa_b(k)} \stackrel{\rm def}{=} \kappa_2(k).$$

Thus (4.16) yields that

$$\begin{aligned}
|\phi(x_k+s_k,\mu_k) - m_{k,j}(x_{k,j}+s_{k,j})| &\leq \tfrac{1}{2}|\langle s_{k,j}, \nabla_{xx}\phi(\xi_{k,j})s_{k,j}\rangle| + \tfrac{1}{2}|\langle s_{k,j}, \nabla_{xx}m_{k,j}(x_{k,j})s_{k,j}\rangle| \\
&\leq \tfrac{1}{2}(\kappa_1(k)+\kappa_2(k))\|s_{k,j}\|_{k,j}^2 \\
&\leq \kappa_\phi(k)\Delta_{k,j}^2,
\end{aligned} \tag{4.17}$$

as required, where we successively used AS.7, the triangle inequality, (2.5) and the fact that $x_{k,j}+s_{k,j} \in \mathcal{B}_{k,j}$ imply that $\|s_{k,j}\|_{k,j} \leq \Delta_{k,j}$. □

We therefore see that the error between the objective function and the model decreases quadratically with the trust-region radius. The smaller this radius becomes, the better the model approximates the objective, which intuitively guarantees that minimizing the model within a sufficiently small trust region will also decrease the objective function, as desired.

We next show that an iteration must be successful if the current iterate is not first-order critical and the trust-region radius is small enough.

Lemma 4.5 Assume that AS.1–AS.8 hold and there exists a $\kappa_g > 0$ such that

$$\|\nabla_x \phi(x_{k,j}, \mu_k)\|_{[k,j]} \geq \kappa_g \tag{4.18}$$

for all j and given k. Then there is a constant $\kappa_\Delta(k) > 0$ only depending on k such that, for all j,

$$\Delta_{k,j} \geq \kappa_\Delta(k).$$

Proof. Assume that iteration ℓ is the first such that

$$\Delta_{k,\ell+1} \leq \gamma_1 \min\left[\kappa_x(k), \frac{\theta\kappa_g(1-\eta_2)}{\max[\kappa_\beta(k), \kappa_\phi(k)]}\right], \tag{4.19}$$

where θ is as in (3.12), $\kappa_x(k)$ is as in Lemma 4.3 and $\kappa_\phi(k)$ is as in Theorem 4.4. But we have from Step 4 of Algorithm 3.1 that $\gamma_1\Delta_{k,\ell} \leq \Delta_{k,\ell+1}$, and hence that

$$\Delta_{k,\ell} \leq \min\left[\kappa_x(k), \frac{\theta\kappa_g(1-\eta_2)}{\max[\kappa_\beta(k), \kappa_\phi(k)]}\right]. \tag{4.20}$$

This latter inequality implies the last part of $\|s_{k,j}\|_{k,\ell} \leq \Delta_{k,\ell} \leq \kappa_x(k)$. Lemma 4.3 now implies that the constraint (3.15) holds and therefore the value of $\phi(x_{k,\ell} + s_{k,\ell}, \mu_k)$ is evaluated. Moreover, since the conditions $\eta_2 \in (0,1)$ and $\theta \in (0, \frac{1}{2})$ imply that $\theta(1-\eta_2) < 1$, we deduce from (4.18) and the bound $\beta_{k,\ell} \leq \kappa_\beta(k)$ that

$$\Delta_{k,\ell} < \frac{\|\nabla_x \phi(x_{k,\ell}, \mu_k)\|_{[k,\ell]}}{\beta_{k,\ell}}.$$

As a consequence, (3.12) and (4.18) immediately give that

$$\begin{aligned} m_\ell(x_{k,\ell}) - m_\ell(x_{k,\ell} + s_\ell) &\geq \theta\|\nabla_x\phi(x_{k,\ell}, \mu_k)\|_{[k,\ell]} \min\left[\frac{\|\nabla_x\phi(x_{k,\ell}, \mu_k)\|_{[k,\ell]}}{\beta_{k,\ell}}, \Delta_{k,\ell}\right] \\ &= \theta\|\nabla_x\phi(x_{k,\ell}, \mu_k)\|_{[k,\ell]}\Delta_{k,\ell} \\ &\geq \theta\kappa_g\Delta_{k,\ell}. \end{aligned}$$

On the other hand, we apply Theorem 4.4 and deduce from this last bound and (4.20) that

$$|\rho_{k,\ell} - 1| = \frac{|\phi(x_{k,\ell} + s_{k,\ell}, \mu_k) - m_{k,\ell}(x_{k,\ell} + s_{k,\ell})|}{|m_{k,\ell}(x_{k,\ell}) - m_{k,\ell}(x_{k,\ell} + s_{k,\ell})|} \leq \frac{\kappa_\phi(k)\Delta_{k,\ell}}{\theta\kappa_g} \leq 1 - \eta_2.$$

Therefore $\rho_{k,\ell} \geq \eta_2$ and $\Delta_{k,\ell+1} \geq \Delta_{k,\ell}$ by Step 4 of Algorithm 3.1. This contradicts our assumption that ℓ is the index of the first iteration at which (4.19) holds. Hence (4.19) is impossible, which yields the desired conclusion with

$$\kappa_\Delta(k) = \gamma_1 \min\left[\kappa_x(k), \frac{\theta\kappa_g(1-\eta_2)}{\max[\kappa_\beta(k), \kappa_\phi(k)]}\right].$$

□

The proof of the convergence of Algorithm 3.1 follows the pattern which is now classical for trust-region methods. We first consider the case where Algorithm 3.1 has only a finite number of successful iterates.

Lemma 4.6 Assume that AS.1–AS.8 hold and that there are only finitely many successful iterates in Algorithm 3.1. Then

$$\|\nabla_x\phi(x_{k,\ell+j}, \mu_k)\|_{[k,\ell+j]} = \|\nabla_x\phi(x_{k,\ell+j}, \mu_k)\|_\diamond = 0,$$

for all $j > 0$, where ℓ is the index of the last successful iteration.

Proof. The mechanism of Algorithm 3.1 ensures that $x_{k,\ell+j}$ remains constant for all $j \geq 0$, where (k,ℓ) is the index of the last successful inner iteration. Moreover, since all inner iterations $(k,\ell+j)$ are unsuccessful for $j > 0$, Step 4 of Algorithm 3.1 implies that $\Delta_{k,\ell+j}$ converges to zero when j tends to infinity. If $\|\nabla_x \phi(x_{k,\ell+j},\mu_k)\|_{[\ell+j]}$ is bounded away from zero, Lemma 4.5 implies that this is also the case for $\Delta_{k,\ell+j}$, which is impossible. The desired conclusion then follows from the fact that all $\|\cdot\|_{[k,j]}$ seminorms are uniformly equivalent to $\|\cdot\|_\diamond$ because of Lemma 4.2. □

If there are infinitely many successful iterations, a similar conclusion holds in the limit, as we now verify.

Lemma 4.7 Assume that AS.1–AS.8 hold and that there are infinitely many successful iterates in Algorithm 3.1. Then

$$\liminf_{j\to\infty} \|\nabla_x \phi(x_{k,j},\mu_k)\|_{[k,j]} = \liminf_{j\to\infty} \|\nabla_x \phi(x_{k,j},\mu_k)\|_\diamond = 0. \tag{4.21}$$

Proof. Assume, for the purpose of deriving a contradiction, that (4.18) holds for all j. Now consider a successful inner iteration (k,ℓ). For this iteration, (3.12), (4.18), (4.5), the inequality $\rho_{k,\ell} \geq \eta_1$ and Lemma 4.5 imply that

$$\phi(x_{k,\ell},\mu_k) - \phi(x_{k,\ell+1},\mu_k) \geq \eta_1[m_{k,\ell}(x_{k,\ell}) - m_{k,\ell}(x_{k,\ell}+s_\ell)] \geq \theta\kappa_g\eta_1 \min\left[\frac{\kappa_g}{\kappa_\beta(k)}, \kappa_\Delta(k)\right] \stackrel{\rm def}{=} \delta_1 > 0.$$

Summing over all successful iterations from 0 to ℓ, we deduce that

$$\phi(x_{k,0},\mu_k) - \phi(x_{k,\ell+1},\mu_k) = {\sum_{j=0}^{\ell}}' [\phi(x_{k,j},\mu_k) - \phi(x_{k,j+1},\mu_k)] \geq \sigma_\ell \delta_1,$$

where the $\sum'$ is restricted to successful iterations and σ_ℓ is the number of successful (inner) iterations from iteration $(k,0)$ up to iteration (k,ℓ). Our assumption then gives that σ_ℓ tends to plus infinity when ℓ grows, and thus we obtain that $\phi(x_{k,\ell+1},\mu_k)$ is unbounded below on $\mathcal{F}$, which contradicts AS.5. Thus (4.18) cannot hold for all ℓ, and the proof is concluded by using Lemma 4.2. □

This result states that at least one limit point of Algorithm 3.1 is first-order critical. We now prove that this property holds for *all* such limit points.

Theorem 4.8 Assume that AS.1–AS.8 hold. Then

$$\lim_{j\to\infty} \|\nabla_x \phi(x_{k,j},\mu_k)\|_{[k,j]} = \lim_{j\to\infty} \|\nabla_x \phi(x_{k,j},\mu_k)\|_\diamond = 0. \tag{4.22}$$

Proof. Lemma 4.6 shows that the conclusion holds if there are only finitely many successful iterations. Assume now that this is not the case, and assume, again for the purpose of obtaining a contradiction, that there is a subsequence of successful (inner) iterates indexed by $\{k,t_i\}$ such that

$$\|\nabla_x \phi(x_{k,t_i},\mu_k)\|_{[k,t_i]} \geq 3\epsilon \tag{4.23}$$

for some $\epsilon > 0$ and for all i. Lemma 4.7 then ensures the existence, for each t_i, of a successful iteration $(k,p(t_i))$ with $p(t_i) > t_i$ such that $\|\nabla_x \phi(x_{k,p(t_i)},\mu_k)\|_{[k,p(t_i)]} < \epsilon$. Denoting $p_i = p(t_i)$, we thus obtain that there exists another subsequence of successful iterates indexed by (k,p_i) such that

$$\|\nabla_x \phi(x_{k,j},\mu_k)\|_{[k,j]} \geq \epsilon \;\text{ for }\; t_i \leq j < p_i \;\text{ and }\; \|\nabla_x \phi(x_{k,p_i},\mu_k)\|_{[k,p_i]} < \epsilon. \tag{4.24}$$

We now restrict our attention to the subsequence of successful iterations whose indices are in the set

$$\mathcal{J} = \{(k,j) \in \mathcal{S} \mid t_i \leq j < p_i\},$$

where t_i and p_i belong to the two subsequences defined above. Using (3.12), the fact that all iterations in $\mathcal{J}$ are successful, (4.5) and (4.24), we deduce that for $(k,j) \in \mathcal{J}$,

$$\phi(x_{k,j},\mu_k) - \phi(x_{k,j+1},\mu_k) \geq \eta_1[m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j}+s_{k,j})] \geq \theta\epsilon\eta_1 \min\left[\frac{\epsilon}{\kappa_\beta(k)}, \Delta_{k,j}\right]. \quad (4.25)$$

But the sequence $\{\phi(x_{k,j},\mu_k)\}_{j=0}^{\infty}$ is monotonically decreasing and bounded below because of AS.5. Hence it is convergent and the left-hand side of (4.25) must tend to zero when j tends to infinity. This gives that

$$\lim_{\substack{j\to\infty \\ (k,j)\in\mathcal{J}}} \Delta_{k,j} = 0.$$

As a consequence, the second term dominates in the minimum of (4.25) and we deduce that, for $(k,j) \in \mathcal{J}$ and j sufficiently large,

$$\Delta_{k,j} \leq \frac{1}{\theta\epsilon\eta_1}[\phi(x_{k,j},\mu_k) - \phi(x_{k,j+1},\mu_k)].$$

We then obtain from this inequality, the observation that $\|x_{k,t_i} - x_{k,p_i}\| = \|x_{k,t_i} - x_{k,p_i}\|_\diamond$ because x_{k,t_i} and x_{k,p_i} both belong to $\mathcal{L}$, and Lemma 4.2 that, for i sufficiently large,

$$\|x_{k,t_i} - x_{k,p_i}\| \leq \kappa_{\rm n}(k) \sum_{j=t_i}^{p_i-1}{}' \|x_{k,j} - x_{k,j+1}\|_{k,j} \leq \kappa_{\rm n}(k) \sum_{j=t_i}^{p_i-1}{}' \Delta_{k,j} \leq \frac{\kappa_{\rm n}(k)}{\theta\epsilon\eta_1}[\phi(x_{k,t_i},\mu_k) - \phi(x_{k,p_i},\mu_k)].$$

Using AS.5 and the monotonicity of the sequence $\{\phi(x_{k,j},\mu_k)\}_{j=0}^{\infty}$ again, we observe that the right-hand side of this last inequality must converge to zero, and therefore that $\|x_{k,t_i} - x_{k,p_i}\|$ tends to zero when i tends to infinity. We then deduce from the continuity of $\nabla_x\phi(x,\mu_k)$ and Lemma 4.2 that

$$\Big| \|\nabla_x\phi(x_{k,t_i},\mu_k)\|_{[k,t_i]} - \|\nabla_x\phi(x_{k,p_i},\mu_k)\|_{[k,p_i]} \Big| \leq \epsilon$$

for i sufficiently large. Using this last bound, (4.23) and (4.24), we then have that

$$\begin{aligned} 2\epsilon &= 3\epsilon - \epsilon \\ &\leq \|\nabla_x\phi(x_{k,t_i},\mu_k)\|_{[k,t_i]} - \|\nabla_x\phi(x_{k,p_i},\mu_k)\|_{[k,p_i]} \\ &\leq \Big| \|\nabla_x\phi(x_{k,t_i},\mu_k)\|_{[k,t_i]} - \|\nabla_x\phi(x_{k,p_i},\mu_k)\|_{[k,p_i]} \Big| \\ &\leq \epsilon, \end{aligned}$$

which is impossible. Hence no subsequence satisfying (4.18) can exist and the theorem is proved. □

This concludes the convergence theory for the inner algorithm, at least as far as convergence to first-order critical points is concerned. However, the tests (3.16)–(3.20) are based on convergence to points satisfying second-order necessary conditions. In order to obtain the necessary results in this direction, we need to strengthen our assumptions on the Hessian of the Lagrangian's model and on the dual variables, as suggested in Section 3.2. More specifically, we assume that,

$\boxed{\textbf{AS.9}}$ for all k,

$$\lim_{j\to\infty} \|G_{k,j} - \nabla_{xx}\psi(x_{k,j},z_{k,j})\|_{\{k,j\}} = 0 \ \text{ when } \ \lim_{j\to\infty} \|\nabla_x\phi(x_{k,j},\mu_k)\|_{[k,j]} = 0,$$

$\boxed{\textbf{AS.10}}$ for all k, $\lim_{j\to\infty} \|z_{k,j} - \mu_k C_{k,j}^{-1} e\| = 0$ when $\lim_{j\to\infty} \|\nabla_x \phi(x_{k,j}, \mu_k)\|_{[k,j]} = 0$.

Note that these two assumptions together imply that

$$\lim_{j\to\infty} \|G_{k,j} - \nabla_{xx} f(x_{k,j}) + \sum_{i=1}^{p} \frac{\mu_k}{c_i(x_{k,j})} \nabla_{xx} c_i(x_{k,j})\|_{\{k,j\}} = 0 \text{ when } \lim_{j\to\infty} \|\nabla_x \phi(x_{k,j}, \mu_k)\|_{[k,j]} = 0. \quad (4.26)$$

We are now in position to prove that the model is asymptotically convex, at least along some subsequence.

Theorem 4.9 Assume that AS.1–AS.10 hold. Then

$$\limsup_{j\to\infty} \lambda^{\mathrm{L}}_{M_{k,j}} [\nabla_{xx} m_{k,j}(x_{k,j})] = \limsup_{j\to\infty} \lambda^{\mathrm{L}}_{M_{k,j}} [G_{k,j} + B_{k,j}] \geq 0 \quad (4.27)$$

and

$$\limsup_{j\to\infty} \lambda^{\mathrm{L}}_{M_{k,j}} [\nabla_{xx} \phi(x_{k,j}, \mu_k)] \geq 0.$$

Proof. Assume first, for the purpose of deriving a contradiction, that there exists an $\epsilon > 0$ such that

$$\lambda^{\mathrm{L}}_{M_{k,j}} [G_{k,j} + B_{k,j}] \leq -\epsilon, \quad (4.28)$$

for all j sufficiently large. Using this definition, (3.12) and (4.22), we then obtain that

$$m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j} + s_{k,j}) \geq \theta\epsilon \min\left[\epsilon^2, \Delta_{k,\ell}^2\right] \geq \theta\epsilon\Delta_{k,j}^2 \quad (4.29)$$

for j sufficiently large and $\Delta_{k,j}$ sufficiently small. We may then again consider the ratio of predicted versus achieved reduction and deduce that, for such j and $\Delta_{k,j}$ and for some $\xi_{k,j}$ in $[x_{k,j}, x_{k,j}+s_{k,j}] \subset \mathcal{F}$ (where the last inclusion holds because of Lemma 4.3),

$$\begin{aligned} |\rho_{k,j} - 1| &= \left| \frac{\phi(x_{k,j} + s_\ell) - m_{k,j}(x_{k,j} + s_{k,j})}{m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j} + s_{k,j})} \right| \\ &\leq \frac{1}{\theta\epsilon\Delta_{k,j}^2} \left[|\langle s_{k,j}, \nabla_{xx}\phi(\xi_{k,j}, \mu_k) s_{k,j}\rangle - \langle s_{k,j}, (G_{k,j} + B_{k,j}) s_{k,j}\rangle| \right] \\ &\leq \frac{1}{\theta\epsilon} \|\nabla_{xx}\phi(\xi_{k,j}, \mu_k) - G_{k,j} - B_{k,j}\|_{\{k,j\}}, \end{aligned} \quad (4.30)$$

where we have used (3.3), (3.6), (2.5) and the bound $\|s_{k,j}\|_{k,j} \leq \Delta_{k,j}$. In order to derive an upper bound on the last right-hand side of this last inequality, we first note that, because of Theorem 4.8 and AS.10,

$$\lim_{j\to\infty} \|z_{k,j} - \mu_k C_{k,j}^{-1} e\| = 0. \quad (4.31)$$

Morever,

$$\|\xi_{k,j} - x_{k,j}\|_{k,j} \leq \|s_{k,j}\|_{k,j} \leq \Delta_{k,j}$$

and we therefore obtain, using (4.31), that

$$\begin{aligned} &\lim_{\Delta_{k,j}\to 0,\, j\to\infty} \|\mu_k J(\xi_{k,j})^T C(\xi_{k,j})^{-2} J(\xi_{k,j}) - B_{k,j}\|_{\{k,j\}} \\ &\quad= \lim_{\Delta_{k,j}\to 0,\, j\to\infty} \|\mu_k J(\xi_{k,j})^T C(\xi_{k,j})^{-2} J(\xi_{k,j}) - J(x_{k,j})^T C(x_{k,j})^{-1} Z_{k,j} J(x_{k,j})\|_{\{k,j\}} \\ &\quad= \mu_k \lim_{\Delta_{k,j}\to 0,\, j\to\infty} \|J(\xi_{k,j})^T C(\xi_{k,j})^{-2} J(\xi_{k,j}) - J(x_{k,j})^T C(x_{k,j})^{-2} J(x_{k,j})\|_{\{k,j\}} \\ &\quad= 0 \end{aligned} \quad (4.32)$$

and also that

$$\lim_{\Delta_{k,j}\to 0} \|\nabla_{xx} f(\xi_{k,j}) - \sum_{i=1}^{p} \frac{\mu_k}{c_i(\xi_{k,j})} \nabla_{xx} c_i(\xi_{k,j}) - \nabla_{xx} f(x_{k,j}) + \sum_{i=1}^{p} \frac{\mu_k}{c_i(x_{k,j})} \nabla_{xx} c_i(x_{k,j})\|_{\{k,j\}} = 0. \quad (4.33)$$

Now observe that, for any $v \in \mathbb{R}^n$,

$$\begin{aligned}
&\|\nabla_{xx}\phi(v,\mu_k) - G_{k,j} - B_{k,j}\|_{\{k,j\}} \\
&\quad \leq \|\nabla_{xx} f(v) - \sum_{i=1}^{p} \frac{\mu_k}{c_i(v)} \nabla_{xx} c_i(v) - \nabla_{xx} f(x_{k,j}) + \sum_{i=1}^{p} \frac{\mu_k}{c_i(x_{k,j})} \nabla_{xx} c_i(x_{k,j})\|_{\{k,j\}} \\
&\quad\quad + \|\nabla_{xx} f(x_{k,j}) - \sum_{i=1}^{p} \frac{\mu_k}{c_i(x_{k,j})} \nabla_{xx} c_i(x_{k,j}) - G_{k,j}\|_{\{k,j\}} \quad (4.34) \\
&\quad\quad + \|\mu_k J(v)^T C(v)^{-2} J(v) - B_{k,j}\|_{\{k,j\}},
\end{aligned}$$

using the definition of the logarithmic barrier function, (3.4), (3.5) and the triangle inequality. Substituting now (4.26), (4.32) and (4.33) in this last inequality with $v = \xi_{k,j}$, we obtain that

$$\lim_{\substack{\Delta_{k,j}\to 0 \\ j\to\infty}} \|\nabla_{xx}\phi(\xi_{k,j},\mu_k) - G_{k,j} - B_{k,j}\|_{\{k,j\}} = 0 \quad (4.35)$$

and thus the last right-hand side of (4.30) is arbitrarily small when j is sufficiently large and $\Delta_{k,j}$ sufficiently small. Thus $\rho_{k,j} \geq \eta_2$ for such j and $\Delta_{k,j}$. Hence there must exist a $\delta_1 \in (0,\epsilon]$ and a $j_0 > 0$ such that

$$\rho_{k,j} \geq \eta_2 \text{ for all } j \geq j_0 \text{ such that } \Delta_{k,j} \leq \delta_1. \quad (4.36)$$

Therefore, each iteration such that this condition hold ensures that $\Delta_{k,j+1} \geq \Delta_{k,j}$ by Algorithm 3.1. This in turn implies that, for $j \geq 0$,

$$\Delta_{k,j_0+j} \geq \min[\gamma_1\delta_1, \Delta_{k,j_0}] \stackrel{\text{def}}{=} \delta_2. \quad (4.37)$$

Combining (4.29) and this lower bound, we obtain that

$$\phi(x_{k,j_0+j},\mu_k) - \phi(x_{k,j_0+j+1},\mu_k) \geq \eta_1\theta\epsilon\delta_2^2 > 0. \quad (4.38)$$

whenever iteration $j_0 + j$ is successful. If there are only finitely many successful iterations, the mechanism of the algorithm implies that the trust-region radius converges to zero, which is impossible because of (4.37). Hence there must be an infinite number of successful iterations. But (4.38) now contradicts AS.5. Hence our assumption (4.28) must be false and (4.27) is proved. The second inequality in the theorem's statement then immediately results from (3.4), (4.34) with $v = x_{k,j}$, (4.26), Theorem 4.8 and (4.31). □

We conclude our analysis of Algorithm 3.1 by returning to the case where the stopping tolerances $\epsilon^{\rm C}$, $\epsilon^{\rm D}$ and $\epsilon^{\rm E}$ are positive instead of being zero, and show that the stopping conditions of Algorithm 3.1 will eventually be satisfied.

Theorem 4.10 Assume that AS.1–AS.10 hold and that

$$\epsilon^{\rm C} > 0, \quad \epsilon^{\rm D} > 0 \text{ and } \epsilon^{\rm E} > 0.$$

Then conditions (3.18)–(3.20) hold after a finite number of iterations of Algorithm 3.1.

Proof. Theorems 4.8 and AS.10 together imply that

$$\|\nabla_x \phi(x_{k,j}, \mu_k)\|_{[k,j]} \to 0 \text{ and } C_{k,j} z_{k,j} - \mu_k e \to 0$$

when j tends to infinity. As a consequence (3.18) and (3.19) both hold after finitely many iterations. Theorem 4.9 then guarantees that (3.20) will also be satisfied eventually, which concludes the proof. □

We note that Theorem 4.9 does not assume that the sequence of iterates of Algorithm 3.1 converges, or even that it has limit points. If this additional assumption is made, then the result may be extended to show that all these limit points satisfy second-order necessary conditions for optimality.

4.3 Updating the Vector of Dual Variables

We now indicate how the dual variables $z_{k,j+1}$ may be updated in practice at Step 5 of the primal-dual barrier algorithm, while ensuring AS.6 and AS.10. A simple idea is to use the value predicted in the middle part of the Newton equations (3.8), which is

$$\overline{z}_{k,j+1} = z_{k,j} + \Delta z_{k,j} = \mu_k C_{k,j}^{-1} e - C_{k,j}^{-1} Z_{k,j} J_{k,j} s_{k,j}. \tag{4.39}$$

However, there is no guarantee that the choice $z_{k,j+1} = \overline{z}_{k,j+1}$ maintains feasibility of the dual variables ($z_{k,j+1} \geq 0$), nor that it satisfies AS.6 or AS.10. We thus need to safeguard it, which can be achieved by projecting (componentwise) the value (4.39) into the interval

$$\mathcal{I} = \left[\kappa_{\rm zl} \min\left(e, z_{k,j}, \mu_k C_{k,j+1}^{-1} e\right), \max\left(\kappa_{\rm zu} e, z_{k,j}, \kappa_{\rm zu} \mu_k^{-1} e, \kappa_{\rm zu} \mu_k C_{k,j+1}^{-1} e\right)\right], \tag{4.40}$$

where $\kappa_{\rm zl}$ and $\kappa_{\rm zu}$ are constants such that

$$0 < \kappa_{\rm zl} < 1 < \kappa_{\rm zu}. \tag{4.41}$$

This is to say that

$$z_{k,j+1} = \begin{cases} P_{\mathcal{I}}[\overline{z}_{k,j+1}] & \text{if } x_{k,j+1} = x_{k,j} + s_{k,j} \\ z_{k,j} & \text{if } x_{k,j+1} = x_{k,j}, \end{cases} \tag{4.42}$$

where $P_{\mathcal{I}}[v]$ is the componentwise projection of the vector v onto the interval $\mathcal{I}$. In practice, $\kappa_{\rm zl} = \frac{1}{2}$ and $\kappa_{\rm zu} = 10^{20}$ appear to work satisfactorily. Does this safeguarded value satisfy the required conditions? We now verify that this is usually the case.

Theorem 4.11 Suppose that AS.2–AS.5 and AS.7–AS.8 hold. Suppose also that $\{x_{k,j}, z_{k,j}\}$ is a sequence of primal and dual iterates generated, at a given outer iteration k, by Algorithm 3.1 where $z_{k,j+1}$ is updated according to (4.42), with $\mathcal{I}$ being given by (4.40) and $\overline{z}_{k,j+1}$ by (4.39). Then $z_{k,j+1} > 0$ and AS.6 holds. If, furthermore,

$$\lim_{j\to\infty} \|s_{k,j}\|_{k,j} = 0 \text{ when } \lim_{j\to\infty} \|\nabla_x \phi(x_{k,j}, \mu_k)\|_{[k,j]} = 0 \tag{4.43}$$

then AS.10 is also satisfied.

Proof. The positivity of the vector of dual variables immediately results from the fact that the lower end of the interval $\mathcal{I}$ is always positive. To obtain AS.6, we notice that the definition of $\mathcal{I}$ and this bound implies that

$$[z_{k,j+1}]_i \leq \max\left[\kappa_{\rm zu}, [z_{k,0}]_i, \frac{\kappa_{\rm zu}}{\mu_k}, \frac{\kappa_{\rm zu}\mu_k}{c_i(x_{k,j+1})}\right]$$

and AS.6 follows with

$$\kappa_{\rm zi}(k) \stackrel{\rm def}{=} \max\left[\kappa_{\rm zu}, [z_{k,0}]_i, \frac{\kappa_{\rm zu}}{\mu_k}, \kappa_{\rm zu}\mu_k\right].$$

We now show that AS.10 is also satisfied if (4.43) holds. Suppose therefore that $\|\nabla_x\phi(x_{k,j},\mu_k)\|_{[k,j]}$ converges to zero, which must eventually occur because of Theorem 4.8. This implies, because of Lemma 4.2, the fact that $As_{k,j}=0$ and (4.43), that

$$\lim_{j\to\infty}\|s_{k,j}\| = \lim_{j\to\infty}\|s_{k,j}\|_\diamond = \lim_{j\to\infty}\|s_{k,j}\|_{k,j} = 0. \tag{4.44}$$

Then Lemma 4.1, (4.44) and AS.2 ensure that

$$\lim_{\substack{j\to\infty\\(k,j)\in\mathcal{S}}}\|C_{k,j}^{-1} - C_{k,j+1}^{-1}\| = 0. \tag{4.45}$$

But

$$\begin{aligned}\|\overline{z}_{k,j+1} - \mu_k C_{k,j+1}^{-1}e\| &\leq \|\overline{z}_{k,j+1} - \mu_k C_{k,j}^{-1}e\| + \mu_k\|(C_{k,j}^{-1} - C_{k,j+1}^{-1})e\| \\ &\leq \|C_{k,j}^{-1}Z_{k,j}J_{k,j}\|\,\|s_{k,j}\| + \mu_k\sqrt{n}\|C_{k,j}^{-1} - C_{k,j+1}^{-1}\|,\end{aligned}$$

where we have used (4.39). We thus obtain from Lemma 4.1, (4.44), (4.4), AS.2 and (4.45) that

$$\lim_{\substack{j\to\infty\\(k,j)\in\mathcal{S}}}\|\overline{z}_{k,j+1} - \mu_k C_{k,j+1}^{-1}e\| = 0.$$

Now this limit and (4.41) give that, for $(k,j)\in\mathcal{S}$ and j sufficiently large,

$$\kappa_{\rm zl}\mu_k C_{k,j+1}^{-1}e \leq \overline{z}_{k,j+1} \leq \kappa_{\rm zu}\mu_k C_{k,j+1}^{-1}e.$$

Hence, from the definition of $z_{k,j+1}$, we have that $z_{k,j+1} = \overline{z}_{k,j+1}$ for $j\in\mathcal{S}$ sufficiently large. Thus (4.39) yields that

$$C_{k,j+1}Z_{k,j+1}e = C_{k,j+1}C_{k,j}^{-1}(-Z_{k,j}J_{k,j}s_{k,j} + \mu_k e). \tag{4.46}$$

On the other hand, we deduce from AS.2, Lemma 4.3 and (4.44) that

$$\lim_{\substack{j\to\infty\\(k,j)\in\mathcal{S}}} C_{k,j+1}C_{k,j}^{-1} = I.$$

We then obtain from this limit, AS.3, AS.6 and (4.46) that

$$\lim_{\substack{j\to\infty\\(k,j)\in\mathcal{S}}} C_{k,j+1}Z_{k,j+1}e = \mu_k e.$$

AS.10 then follows because $z_{k,j+1} = z_{k,j}$ for $(k,j)\notin\mathcal{S}$, that is exactly when $x_{k,j+1} = x_{k,j}$. □

Observe that the first part of the proof implies that any value of $z_{k,j+1}$ chosen in $\mathcal{I}$ satisfies AS.6. In particular, this is true for the choices

$$z_{k,j+1} = z_{k,j} \;\text{ and }\; z_{k,j+1} = \mu_k C_{k,j+1}^{-1}e,$$

the latter corresponding to the pure primal method, that is to the model (3.3). Also note that, because of Theorem 4.2, the choice of norms in (4.43) is in fact irrelevant : the Euclidean norm would have been just as adequate, but we have chosen the scaled norms for consistency.

4.4 Convergence of the outer iteration

Having proved that its iterates are well-defined, we now consider the convergence of Algorithm 3.2. In order to state our result, we need the following definition. We say that a subsequence of outer iterates $\{x_{k_\ell}\}$ is consistently active if, for each $i = 1, \ldots, p$ either

$$\lim_{\ell\to\infty} c_i(x_{k_\ell}) = 0 \quad \text{or} \quad \liminf_{\ell\to\infty} c_i(x_{k_\ell}) > 0.$$

This is to say that each constraint is asymptotically active or inactive for the complete subsequence. We also define the set of asymptotically active constraints for such a subsequence by

$$\mathcal{A}\{x_{k_\ell}\} \stackrel{\text{def}}{=} \{i \in \{1, \ldots, n\} \mid \lim_{\ell\to\infty} c_i(x_{k_\ell}) = 0\}.$$

In other words, the set of asymptotically active constraints is fixed for the iterates of a consistently active subsequence. Since there are only a finite number of such sets, as each constraint is asymptotically active or is not, the number of consistently active subsequences is finite for any sequence $\{x_k\}$ of non-negative iterates. Furthermore, the complete sequence of iterates may be partitioned into disjoint consistently active subsequences. Observe also that, if $\{x_k\}$ has limit points, then each subsequence converging to a specific limit point x_* is consistently active, as the set of asymptotically active constraints is then determined by the components of x_*, that is $\mathcal{A}\{x_{k_\ell}\} = \{i \in \{1, \ldots, n\} \mid c_i(x_*) = 0\}$.

We then have the following result.

Theorem 4.12 Suppose that AS.1–AS.10 hold. Suppose also that, for some $\kappa_\mu > 0$,

$$\lim_{k\to\infty} \frac{\epsilon^{\text{C}}(\mu_k)}{\mu_k} \leq \kappa_\mu, \tag{4.47}$$

that

$$\lim_{k\to\infty} \frac{\epsilon^{\text{D}}(\mu_k)\sqrt{\mu_k}}{\min_i c_i(x_{k+1})} = 0 \tag{4.48}$$

and that $\{x_k\}$ is a sequence of iterates generated by Algorithm 3.2. Then, we have that

$$\lim_{k\to\infty} [N^T \nabla_x f(x_k)]_i - [N^T J_k^T z_k]_i = 0, \quad (i = 1, \ldots, m). \tag{4.49}$$

Furthermore, we also have that, for every consistently active subsequence of iterates $\{x_{k_\ell}\}$,

$$\lim_{\ell\to\infty} [z_{k_\ell}]_i = 0, \quad (i \notin \mathcal{A}\{x_{k_\ell}\}) \tag{4.50}$$

and

$$\liminf_{\ell\to\infty} \langle u_{k_\ell}, N^T \nabla_{xx}\psi(x_{k_\ell}) N u_{k_\ell}\rangle \geq 0 \tag{4.51}$$

for every sequence $\{u_{k_\ell}\}$ in $\mathbb{R}^m$ for which $[J_{k_\ell} N u_{k_\ell}]_i = 0$ whenever $i \in \mathcal{A}\{x_{k_\ell}\}$.

Proof. We start by choosing a subsequence of $\{x_k\}$ indexed by $\mathcal{K}$ such that

$$\lim_{k\to\infty} \frac{[z_k]_i}{c_i(x_k)} = +\infty \quad (i \in \mathcal{E}) \qquad \text{and} \qquad \limsup_{k\to\infty} \frac{[z_k]_i}{c_i(x_k)} < \infty \quad (i \in \mathcal{R}), \tag{4.52}$$

for some subsets $\mathcal{E}$ and $\mathcal{R}$ of $\{1, \ldots, p\}$. The contraints whose index is in $\mathcal{E}$ converge quickly to zero (they are "eager"), while those whose index is in $\mathcal{R}$ are "reluctant" to do so, if they converge to zero at all. Note that the complete sequence of iterates may again be partitioned into a finite set of subsequences satisfying (4.52) (for different sets $\mathcal{E}$ and $\mathcal{R}$). Let $\kappa_3 > 0$ be such that

$$\kappa_3 \geq \max_{i \in \mathcal{R}} \limsup_{\substack{k \to \infty \\ k \in \mathcal{K}}} \frac{[z_k]_i}{c_i(x_k)}.$$

Writing $r_k = N^T[\nabla_x f(x_k) - J_k^T Z_k e]$ and using (2.3), the definition of the k-seminorm, condition (3.19) then becomes

$$\|(N^T M_k N)^{-\frac{1}{2}} r_k\| \leq \epsilon^{\mathrm{D}}(\mu_{k-1}) \tag{4.53}$$

for all k. But, since $N^T M_k N$ is positive definite,

$$\|(N^T M_k N)^{-\frac{1}{2}} r_k\|^2 \geq \frac{\|r_k\|^2}{\lambda^{\mathrm{R}}[N^T M_k N]} = \frac{\|r_k\|^2}{\|N^T M_k N\|}.$$

Now, we have, using (4.2), that, for $k \in \mathcal{K}$ sufficiently large,

$$\|N^T M_k N\| \leq \|N^T W_k N\| + \|N^T B_k N\| \leq \kappa_{\mathrm{W}} + \|J_k^T C_k^{-1} Z_k J_k\| = \kappa_{\mathrm{W}} + \kappa_J^2 \max_i \frac{[z_k]_i}{c_i(x_k)}.$$

Assume first that $\mathcal{E} = \emptyset$. Then,

$$\|N^T M_k N\| \leq \kappa_{\mathrm{W}} + 2\kappa_J^2 \kappa_3$$

for $k \in \mathcal{K}$ sufficiently large, and therefore, using (3.19) and (4.53),

$$\epsilon^{\mathrm{D}}(\mu_{k-1}) \geq \|(N^T M_k N)^{-\frac{1}{2}} r_k\| \geq \frac{\|r_k\|}{\sqrt{\kappa_{\mathrm{W}} + 2\kappa_J^2 \kappa_3}}$$

for such k. This implies that

$$\lim_{\substack{k \to \infty \\ k \in \mathcal{K}}} \|r_k\| = \lim_{\substack{k \to \infty \\ k \in \mathcal{K}}} \|N^T \nabla_x f(x_k) - N^T J_k^T z_k\| = 0. \tag{4.54}$$

On the other hand, if $\mathcal{E} \neq \emptyset$, we first observe that, for each i,

$$\frac{[z_k]_i}{c_i(x_k)} \leq \frac{\mu_{k-1}}{c_i(x_k)^2} + \frac{|c_i(x_k)[z_k]_i - \mu_{k-1}|}{c_i(x_k)^2} \leq \frac{\mu_{k-1}}{c_i(x_k)^2} + \frac{\epsilon^{\mathrm{C}}(\mu_{k-1})}{c_i(x_k)^2} \leq (1 + \kappa_\mu) \frac{\mu_{k-1}}{c_i(x_k)^2},$$

where we have used the triangle inequality, (3.18) and (4.47) successively. Thus we obtain that

$$\|N^T M_k N\| \leq 2\kappa_J^2 \max_i \frac{[z_k]_i}{c_i(x_k)} \leq 2(1 + \kappa_\mu)\kappa_J^2 \frac{\mu_{k-1}}{\min_i c_i(x_k)^2} \stackrel{\mathrm{def}}{=} \kappa_4^2 \frac{\mu_{k-1}}{\min_i c_i(x_k)^2}$$

for $k \in \mathcal{K}$ sufficiently large. In this case,

$$\|(N^T M_k N)^{-\frac{1}{2}} r_k\| \geq \|r_k\| \frac{\min_i c_i(x_k)}{\kappa_4 \sqrt{\mu_{k-1}}}$$

and hence, using (4.53),

$$\|r_k\| \leq \kappa_4 \frac{\epsilon^{\mathrm{D}}(\mu_{k-1})\sqrt{\mu_{k-1}}}{\min_i c_i(x_k)},$$

which, together with (4.48), again yields (4.54). Thus (4.49) holds since $\mathcal{K}$ was chosen arbitrarily.

Suppose now that $\{x_{k_\ell}\}$ is a consistently active subsequence whose set of asymptotically active constraints is $\mathcal{A}$. Then, if $i \notin \mathcal{A}$, (3.18) yields (4.50).

The final step of our proof is to show (4.51), that is that the Hessian of the Lagrangian is, along a consistently active subsequence, asymptotically positive semi-definite in the plane tangent to the asymptotically active constraints. We first notice that (3.20), the forcing nature of $\epsilon^{\mathrm{E}}(\mu)$ and the convergence of μ_k to zero implies that

$$\liminf_{j\to\infty} \inf_{\substack{v\neq 0\\ v\in\mathbb{R}^m}} \frac{\langle v, (N^T M_{k_\ell} N)^{-\frac{1}{2}} N^T [G_{k_\ell} + B_{k_\ell}] N (N^T M_{k_\ell} N)^{-\frac{1}{2}} v\rangle}{\|v\|^2} \geq 0.$$

Hence we deduce from (3.18), AS.9, (3.19) and the convergence of μ_k to zero that

$$\liminf_{j\to\infty} \inf_{\substack{v\neq 0\\ v\in\mathbb{R}^m}} \frac{\langle v, (N^T M_{k_\ell} N)^{-\frac{1}{2}} N^T [\nabla_{xx}\psi(x_{k_\ell}, z_{k_\ell}) + B_{k_\ell}] N (N^T M_{k_\ell} N)^{-\frac{1}{2}} v\rangle}{\|v\|^2} \geq 0.$$

Thus, if we define $w = (N^T M_{k_\ell} N)^{-\frac{1}{2}} v \in \mathbb{R}^m$, we obtain that

$$\begin{aligned}
&\liminf_{\ell\to\infty} \inf_{w\neq 0} \frac{\langle w, N^T [\nabla_{xx}\psi(x_{k_\ell}, z_{k_\ell}) + B_{k_\ell}] N w\rangle}{\|w\|^2_{N^T M_{k_\ell} N}} \\
&= \liminf_{\ell\to\infty} \inf_{v\neq 0} \frac{\langle v, (N^T M_{k_\ell} N)^{-\frac{1}{2}} N^T [\nabla_{xx}\psi(x_{k_\ell}, z_{k_\ell}) + B_{k_\ell}] N (N^T M_{k_\ell} N)^{-\frac{1}{2}} v\rangle}{\|v\|^2} \geq 0,
\end{aligned} \tag{4.55}$$

where we have used the identity $\|w\|_{N^T M_{k_\ell} N} = \|(N^T M_{k_\ell} N)^{\frac{1}{2}} w\| = \|v\|$.

We now assume that (4.51) does not hold, which means that we can pick a sequence of unit vectors $\{u_{k_{\ell_t}}\}$ and a subsequence $\{x_{k_{\ell_t}}\} \subseteq \{x_{k_\ell}\}$ such that

$$[J_{k_{\ell_t}} N u_{k_{\ell_t}}]_i = 0 \quad \text{for} \quad i \in \mathcal{A} \quad \text{and} \quad \liminf_{t\to\infty} \langle u_{k_{\ell_t}}, N^T \nabla_{xx}\psi(x_{k_{\ell_t}}, z_{k_{\ell_t}}) N u_{k_{\ell_t}}\rangle = -\epsilon \tag{4.56}$$

for some $\epsilon > 0$. Using the first part of (4.56), (3.18), (3.4), the convergence of μ_k to zero, AS.10 and the fact that $c_i(x_{k_{\ell_t}})$ is bounded away from zero for $i \notin \mathcal{A}$, we now observe that

$$\lim_{t\to\infty} \langle u_{k_{\ell_t}}, N^T B_{k_{\ell_t}} N u_{k_{\ell_t}}\rangle = \lim_{\ell\to\infty} \mu_{k_{\ell_t}-1} \langle u_{k_{\ell_t}}, N^T J^T_{k_{\ell_t}} C^{-2}_{k_{\ell_t}} J_{k\ell_t} N u_{k_{\ell_t}}\rangle = 0,$$

and hence, taking (4.2) into account, that

$$\|u_{k_{\ell_t}}\|^2_{N^T M_{k_\ell} N} = \langle u_{k_{\ell_t}}, N^T W_{k_\ell} N u_{k_{\ell_t}}\rangle + \langle u_{k_{\ell_t}}, N^T B_{k_{\ell_t}} N u_{k_{\ell_t}}\rangle \leq 2\kappa_{\mathrm{W}}$$

for t sufficiently large. Combining these conclusions, we obtain that

$$\begin{aligned}
-\epsilon &= \liminf_{t\to\infty} \langle u_{k_{\ell_t}}, N^T \nabla_{xx}\psi(x_{k_{\ell_t}}, z_{k_{\ell_t}}) N u_{k_{\ell_t}}\rangle \\
&= \liminf_{t\to\infty} \langle u_{k_{\ell_t}}, N^T \nabla_{xx}\psi(x_{k_{\ell_t}}, z_{k_{\ell_t}}) N u_{k_{\ell_t}}\rangle + \liminf_{t\to\infty} \langle u_{k_{\ell_t}}, N^T B_{k_{\ell_t}} N u_{k_{\ell_t}}\rangle \\
&= \liminf_{t\to\infty} \langle u_{k_{\ell_t}}, N^T (\nabla_{xx}\psi(x_{k_{\ell_t}}, z_{k_{\ell_t}}) + B_{k_{\ell_t}}) N u_{k_{\ell_t}}\rangle \\
&\geq 2\kappa_{\mathrm{W}} \liminf_{t\to\infty} \frac{\langle u_{k_{\ell_t}}, N^T (\nabla_{xx}\psi(x_{k_{\ell_t}}, z_{k_{\ell_t}}) + B_{k_{\ell_t}}) N u_{k_{\ell_t}}\rangle}{\|u_{u_{\ell_t}}\|^2_{N^T M_{k_{\ell_t}} N}} \\
&\geq 0,
\end{aligned}$$

where we used (4.55) to obtain the last inequality. This is impossible since $\epsilon > 0$. Hence no vector satisfying (4.56) can exist, (4.51) holds and the proof of the theorem is complete. □

Necessary optimality conditions for (1.1) are that the primal variables x^* and dual variables z^* satisfy the first-order optimality conditions

$$Ax^* = b, \quad (x^*, z^*) \geq 0, \quad C(x^*) Z^* = 0 \quad \text{and} \quad N^T(g(x^*) - J(x^*)^T z^*) = 0, \tag{4.57}$$

and the second-order conditions

$$\langle s, \nabla_{xx}\psi(x^*, z^*) s\rangle \geq 0 \quad \text{for all} \quad s \in \mathcal{U}, \tag{4.58}$$

where

$$\mathcal{U} = \left\{ s \;\middle|\; \begin{array}{l} As = 0, \\ [J(x^*)s]_i = 0 \quad \text{if} \;\; c_i(x^*) = 0 \end{array} \right\} \tag{4.59}$$

(see, for example, Gill, Murray and Wright, 1981, page 81). This definition of $\mathcal{U}$ corresponds to the "weak" second-order necessary conditions. Ideally, we would like to obtain their "strong" counterpart, in which (4.58) holds for

$$\mathcal{U} = \left\{ s \;\middle|\; \begin{array}{c} As = 0, \\ [J(x^*)s]_i = 0 \quad \text{if} \;\; c_i(x^*) = 0 \quad \text{and} \quad [z^*]_i > 0, \quad \text{and} \\ [J(x^*)s]_i \geq 0 \quad \text{if} \;\; c_i(x^*) = 0 \quad \text{and} \quad [z^*]_i = 0 \end{array} \right\} \tag{4.60}$$

(see, for example, Fletcher, 1981, Sections 9.2 and 9.3), but we know from Gould and Toint (1999) that this is in general impossible in the framework of log-barrier functions. Thus every limit point of Algorithm 3.2 is first-order critical and satisfies second-order conditions that are as strong as can reasonably be expected.

We conclude our analysis by commenting on condition (4.48). Since Mifflin (1975) has shown that, under reasonable conditions, the quantity $\min_i c_i(x_{k+1})$ is of the order of μ_k or $\mu_k^{\frac{1}{2}}$ depending on strict complementarity, we may then deduce that requiring that $\epsilon^{\mathrm{C}}(\mu)$ and $\epsilon^{\mathrm{D}}(\mu)$ converge to zero faster than μ (which is our choice in the next section) is usually sufficient in practice to ensure convergence of the outer iteration. However, a stopping rule based on (4.48) might be preferable especially when the Jacobian of the contraints is (asymptotically) rank deficient.

5 Numerical Experience

Although the algorithm we have developed in this paper is intended for problems with linear equality constraints and general inequality constraints, to date we have only tested it on the narrower class of non-convex quadratic programming (QP) problems. This was quite deliberate since we have a large number of test examples in this case, and since we already have numerical results for these examples using other QP algorithms. We view non-convex QP as prototypical linearly constrained optimization problems, and thus we hope to see that our new algorithm is effective in at least this case. Furthermore, such problems occur both in their own right, and as subproblems within algorithms for more general constrained optimization.

`VE12` is the new primal-dual non-convex QP Fortran 90 package from the Harwell Subroutine Library (HSL). It is exactly the algorithm we analysed in this paper (specialized to the case of a quadratic objective function), but of course in addition there are a large number of linear algebra tricks and other issues to enhance efficiency. General simple bounds $l \leq x \leq u$ are allowed with some/all of l or u being infinite. All fixed variables are removed automatically and the minimization performed with respect to the remaining variables. The resulting trust-region subproblem (3.11) is (approximately) solved using the generalized Lanczos trust-region (GLTR) algorithm proposed by Gould et al. (1999) and implemented within the HSL as `VF05`. This method was originally proposed for unconstrained problems, but the extra requirement $As = 0$ is imposed via the preconditioner. That is, letting $M = M_{k,j}$, the basic preconditioning step requires the solution of the system

$$\begin{pmatrix} M & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} s^i \\ y^i \end{pmatrix} = - \begin{pmatrix} g^i \\ 0 \end{pmatrix} \tag{5.1}$$

to find a correction s^i, given the gradient g^i of the model at the i-th GLTR iteration—some form of iterative refinement or residual adjustment is needed to ensure that the condition $As = 0$ is satisfied very accurately (see Gould, Hribar and Nocedal, 1998). `VE12` offers the option of a large variety of preconditioners of the form

$$K = \begin{pmatrix} M & A^T \\ A & 0 \end{pmatrix},$$

where M varies from the simplest ($M = I$) to the exact form ($M = H + X^{-1}Z$). However, M is required to be second-order sufficient, and this is enforced by factorizing K and, if K has more than $\mathrm{rank}(A)$ negative eigenvalues, adding $\|M\|$ to M and re-factorizing K. While such a modification strategy is undoubtedly simplistic, it has been effective in our experiments.

The results we present here were obtained using an "automatic" preconditioning strategy that we will now describe We start with just a diagonal Hessian based on the barrier terms, i.e, $M = X^{-1}Z$. This is often sufficient, but if the CPU time per iteration seems to be increasing significantly, we switch to a full factorization $M = H + X^{-1}Z$ for the next iteration. If the cost of this is much higher, we revert to the original preconditioner until the cost again rises to the (now known) value of the full factorization. Of course, we might conceive of adding other levels of preconditioner, but the above seems to perform adequately in most cases. Two other points are important. Firstly, if the model Hessian is itself diagonal, then this is used at every stage. Secondly, if M is diagonal (and nonsingular) and so long as the constraints do not have columns with more than (in our case) 10 nonzeros, we solve the normal equations

$$s = -M^{-1}(A^T y^i + g^i), \quad \text{where} \quad AM^{-1}A^T y^i = -AM^{-1}g^i, \tag{5.2}$$

using the factors of $AM^{-1}A^T$, rather than solving the augmented system (5.1). At some stage we intend to handle denser columns and zero diagonal terms in the normal equation case.

The initial strictly feasible point is found as the analytic centre of the feasible region using another new HSL code, `LA07`. This code is based on the primal-dual infeasible interior point algorithm considered by Conn et al. (1999), and is used in the special case where the objective function is absent. In principle, any good interior-point method would suffice, but in any event, this part of the calculation is usually very efficient.

The initial dual variables z_0 are simply those calculated at the analytic centre x_0, while the initial value of the barrier parameter is the smallest power of 10 larger than $\langle z_0, x_0 \rangle / n$. The barrier parameter is updated so that

$$\mu_{k+1} = \min(0.1\mu_k, \mu_k^{1.5})$$

with the intention of encouraging asymptotic superlinear convergence. The forcing functions which control the inner-iteration convergence are defined to be

$$\epsilon^{\mathrm{C}}(\mu) = \epsilon^{\mathrm{D}}(\mu) = \mu^{1.01} = -\epsilon^{\mathrm{E}}(\mu).$$

The algorithm is halted as soon as an inner-iteration has been terminated with each of these tolerances below 0.0001, or if more than 1000 iterations have been performed. Values $\eta_1 = 0.01$, $\eta_2 = 0.9$ are used to accept and reject steps in the inner-iteration, and the trust-region is updated according to the usual rule

$$\Delta_{k,j+1} = \begin{cases} \min[10^{20}, \max(2\|s_{k,j}\|_{k,j}, \Delta_{k,j})] & \text{if } \rho_{k,j} \geq \eta_2, \\ \Delta_{k,j} & \text{if } \rho_{k,j} \in [\eta_1, \eta_2), \\ \frac{1}{2}\Delta_{k,j} & \text{if } \rho_{k,j} < \eta_1; \end{cases}$$

the initial radius for each inner iteration is $\Delta_{k,0} = 1000\mu_k$.

To test our algorithm, we have selected all of the larger quadratic programs in the CUTE test set (see, Bongartz, Conn, Gould and Toint, 1995). Although it is desirable in practice to preprocess the problems (for instance, to remove redundant constraints and scale the problem, see for example Andersen, Gondzio, Mészáros and Xu, 1996), we have not done so.

In Tables 5.1–5.3, we give the results of our preliminary tests. They were performed in double precision on an IBM RISC System/6000 3BT workstation with 64 Megabytes of RAM, using the xlf90 compiler and optimization level -O3. For each example, we report its name along with its dimensions (n is the number of variables, m the number of constraints), the problem type (C for convex, SOS for second-order sufficient and NC for non-convex and not second-order sufficient), the number of iterations performed (its), and the time taken in seconds (time). For comparison, the tables also show the number of iterations and time taken by a Fortran 90 version of `VE09`, a quadratic programming subroutine from the HSL. This latter algorithm is designed to handle non-convex problems and is of the active-set type, each of its iterations corresponding to a pivoting operation. The reader is referred to Gould (1991) for further details on this method. Note that since iterations mean completely different things for the two approaches, they are not directly comparable, and we include them simply for information. All runs were terminated after 1800 seconds, and any exceeding this limit may be regarded as failures.

In Table 5.1, we report results for what are, by today's standards, relatively small problems. We indicate the better of the two CPU times for each problem in bold. Observe that in the majority of cases the new algorithm outperforms its active-set rival, and that the algorithm is just as successful when the problem is non-convex as it is in the convex case. Such behaviour is at variance with our previous linesearch-based primal-dual method (see Conn et al., 1999) which was far less successful in the non-convex case. We believe that this is likely because negative curvature is better handled in the trust-region subproblem than through the ad-hoc matrix modification strategy which lays at the heart of our previous linesearch algorithm. Of course, the new algorithm is not uniformly better than `VE09`; the `PRIMAL*` and `DUAL*` problems, which require very few changes of active-set, and the `QP*` problems, which need a relatively large number of primal-dual iterations, being cases in point. In addition, `VE12` is currently unable to cope with rank-deficient A, and we are presently investigating the best ways of dealing with this defect.

				VE12		VE09	
Name	n	m	type	its	time	its	time
AUG2DCQP	3280	1600	C	25	**6**	3112	133
AUG2DQP	3280	1600	C	30	**7**	3019	127
AUG3DCQP	3873	1000	C	23	**9**	3056	106
AUG3DQP	3873	1000	C	24	**9**	2097	71
BLOCKQP1	2006	1001	NC	23	**10**	1006	28
BLOCKQP2	2006	1001	NC	29	**8**	1006	40
BLOCKQP3	2006	1001	NC	157	46	1006	**28**
BLOWEYA	2002	1002	C	7	**3**	1597	68
BLOWEYB	2002	1002	C	8	**2**	1497	67
BLOWEYC	2002	1002	C	5	**3**	1697	53
CVXQP1	1000	500	C	39	**35**	861	70
CVXQP2	1000	250	C	37	12	370	13
CVXQP3	1000	750	C	89	**41**	1389	107
DUALC1	223	215	C	35	1	12	**0**
DUALC2	235	229	C	28	1	14	**0**
DUALC5	285	278	C	17	1	10	**0**
DUALC8	510	503	C	25	2	11	**0**
GOULDQP2	699	349	C	3	**0**	251	1
GOULDQP3	699	349	C	10	**0**	463	2
KSIP	1021	1001	C	30	**7**	1388	36
MOSARQP1	1500	600	C	50	**8**	5859	91
MOSARQP2	1500	600	C	43	**7**	1679	27
NCVXQP1	1000	500	NC	76	**5**	1561	51
NCVXQP2	1000	500	NC	66	**4**	1840	61
NCVXQP3	1000	500	NC	112	**17**	too ill-cond. basis	
NCVXQP4	1000	250	NC	48	**1**	649	2
NCVXQP5	1000	250	NC	42	**1**	565	2
NCVXQP6	1000	250	NC	59	10	532	**3**
NCVXQP8	1000	750	NC	49	**6**	1901	141
NCVXQP7	1000	750	NC	56	**6**	1567	120
NCVXQP9	1000	750	NC	75	**22**	too ill-cond. basis	
PRIMALC1	239	9	C	130	1	20	**0**
PRIMALC2	238	7	C	28	4	4	**0**
PRIMALC5	295	8	C	100	1	14	**0**
PRIMALC8	528	8	C	129	128	20	**0**
PRIMAL1	410	85	C	31	4	361	4
PRIMAL2	745	96	C	35	**6**	677	12

PRIMAL3	856	111	C	37	**27**	798	35
PRIMAL4	1564	75	C	27	**18**	1515	40
QPCBOEI1	726	351	C	87	9	823	**6**
QPCBOEI2	305	166	C	81	3	303	**1**
QPCSTAIR	614	356	C	222	18	987	**16**
QPNBOEI1	726	351	NC	$>$ 1000	132	736	**5**
QPNBOEI2	305	166	NC	165	7	299	**1**
QPNSTAIR	614	356	NC	300	38	993	**15**
SOSQP1	2000	1001	SOS	10	**2**	996	14
STCQP1	4097	2052	NC	A rank deficient		2845	**67**
STCQP2	4097	2052	NC	22	**81**	2040	98
STNQP1	4097	2052	NC	A rank deficient		3158	**68**
STNQP2	4097	2052	NC	25	**1**	1408	39
UBH1	909	600	C	5	**0**	315	5
YAO	1002	500	C	72	3	3	**2**

Table 5.1: Preliminary numerical results: small problems

In Tables 5.2 and 5.3, we exhibit specimen results for medium and large-scale instances of the variable-dimensional problems. We include these simply to show that the advantages of interior-point methods over conventional active-set approaches are now most obvious.

				VE12		VE09	
Name	n	m	type	its	time	its	time
AUG2DCQP	20200	10000	C	31	**69**	-	>1800
AUG2DQP	20200	10000	C	35	**77**	-	>1800
AUG3DCQP	27543	8000	C	35	**744**	-	>1800
AUG3DQP	27543	8000	C	26	**598**	-	>1800
BLOCKQP1	20006	10001	NC	26	**673**	-	>1800
BLOWEYB	20002	10002	C	7	**48**	5156	893
CVXQP3	15000	11250	C	24	**104**	-	>1800
GOULDQP2	19999	9999	C	1	**1**	-	>1800
GOULDQP3	19999	9999	C	1	**2**	1331	730
KSIP	10021	10001	C	32	**110**	-	>1800
MOSARQP1	30000	10000	C	57	**455**	not enough memory	
NCVXQP4	10000	2500	NC	53	**33**	6588	343
SOSQP1	20000	10001	SOS	7	**54**	9996	1551
STCQP1	8193	4095	NC	A rank deficient		5769	**268**
STCQP2	8193	4095	NC	18	**246**	4320	613
UBH1	18009	12000	C	5	**13**	-	>1800
YAO	20002	10000	C	107	**118**	not enough memory	

Table 5.2: Preliminary numerical results: specimen medium problems

				VE12		VE09	
Name	n	m	type	its	time	its	time
GOULDQP2	100001	50000	C	3	**32**	-	>1800
GOULDQP3	100001	50000	C	10	**98**	-	>1800

Table 5.3: Preliminary numerical results: specimen large problems

We cannot give results for our other variable dimensional problems in the large category (say 10^5 variables) simply because we do not have enough memory to form the factors of the preconditioner. Clearly, this indicates some limitations of our approach, but since we are able to report successful results for larger problems than we have seen before, we believe that this is an indication that our approach is an important advance in the methods for the numerical solution of large-scale non-convex quadratic programs, with, hopefully, implications for general nonlinear problems.

6 Conclusion

We have introduced a primal-dual algorithm for solving nonlinear non-convex mathematical programming problems with linear equality constraints and general nonlinear inequality constraints. In this algorithm, a scaled trust-region subproblem is approximately solved. Additionally, we have shown that this algorithm is globally convergent to points satisfying the weak second-order necessary optimality conditions, even if we allow the scaling matrices to become unbounded to reflect the singularity of the barrier. Preliminary numerical experiments on a variety on convex and non-convex quadratic programs indicate that the new algorithm is potentially efficient for the solution of large-scale problems.

The analysis presented here can still be extended in several directions. For instance, it is possible to verify that we can replace the quadratic models of the objective function and inequality constraints by more general models, provided they agree with the modelled function at least to first order and have bounded second derivatives. The extension to general nonlinear equality constraints, although less direct, is also worth investigating.

Acknowledgements

The authors are grateful to R. Byrd, J. Nocedal and A. Sartenaer for their interest and their useful comments.

References

E. D. Andersen, J. Gondzio, C. Mészáros, and X. Xu. Implementation of interior point methods for large scale linear programming. *In* T. Terlaky, ed., 'Interior Point Methods in Mathematical Programming', pp. 189–252, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

A. S. El Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang. On the formulation and theory of Newton interior point methods for nonlinear programming. *Journal of Optimization Theory and Applications*, **89**(3), 507–541, 1996.

I. Bongartz, A. R. Conn, N. I. M. Gould, and Ph. L. Toint. CUTE: Constrained and Unconstrained Testing Environment. *Transactions of the ACM on Mathematical Software*, **21**(1), 123–160, 1995.

R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large scale nonlinear programming. Technical Report OTC 97/05, Optimization Technology Center, Northwestern University, Evanston, Illinois, USA, 1997.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. A primal-dual algorithm for minimizing a non-convex function subject to bound and linear equality constraints. *In* G. Di Pillo and F. Giannessi, eds., 'Nonlinear Optimization and Applications 2'. To appear, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.

J.-P. Dussault. Numerical stability and efficiency of penalty algorithms. *SIAM Journal on Numerical Analysis*, **32**(1), 296–317, 1995.

R. Fletcher. *Practical Methods of Optimization: Constrained Optimization.* J. Wiley and Sons, Chichester and New York, 1981.

A. Forsgren and P. E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM Journal on Optimization*, **8**(4), 1132–1152, 1998.

D. M. Gay, M. L. Overton, and M. H. Wright. A primal-dual interior method for nonconvex nonlinear programming. *In* Y. Yuan, ed., 'Proceedings of the 1996 International Conference on Nonlinear Programming, Beijing, China', Kluwer Academic Publishers, Dordrecht, The Netherlands, 31–56, 1998.

P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization.* Academic Press, London and New York, 1981.

N. I. M. Gould. On practical conditions for the existence and uniqueness of solutions to the general equality quadratic-programming problem. *Mathematical Programming*, **32**(1), 90–99, 1985.

N. I. M. Gould. On the convergence of a sequential penalty function method for constrained minimization. *SIAM Journal on Numerical Analysis*, **26**(1), 107–128, 1989.

N. I. M. Gould. An algorithm for large-scale quadratic programming. *IMA Journal of Numerical Analysis*, **11**(3), 299–324, 1991.

N. I. M. Gould, M. E. Hribar, and J. Nocedal. On the solution of equality constrained quadratic problems arising in optimization. Technical Report RAL-TR-98-069, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 1998.

N. I. M. Gould, S. Lucidi, M. Roma, and Ph. L. Toint. Solving the trust-region subproblem using the Lanczos method. *SIAM Journal on Optimization*, **9**(2), 504–525, 1999.

N. I. M. Gould and Ph. L. Toint. A Note on the Second-order Convergence of Optimization Algorithms Using Barrier Functions. *Mathematical Programming*, **85**(2), 433–438, 1999.

R. Mifflin. Convergence bounds for nonlinear programming algorithms. *Mathematical Programming*, **8**(3), 251–271, 1975.

R. J. Vanderbei and D. F. Shanno. An interior point algorithm for nonconvex nonlinear programming. Technical Report SOR 97-21, Program in Statistics and Operations Research, Princeton University, New Jersey, USA, 1997.

H. Yamashita, H. Yabe, and T. Tanabe. A globally and superlinearly convergent primal-dual interior point trust region method for large scale constrained optimization. Technical report, Mathematical Systems Inc., Shinjuku-ku, Tokyo, Japan, July 1997.