SIAM J. SCI. COMPUT.
Vol. 30, No. 4, pp. 1706–1729

GLOBAL AND FINITE TERMINATION OF A TWO-PHASE AUGMENTED LAGRANGIAN FILTER METHOD FOR GENERAL QUADRATIC PROGRAMS*

MICHAEL P. FRIEDLANDER† AND SVEN LEYFFER‡

Abstract. We present a two-phase algorithm for solving large-scale quadratic programs (QPs). In the first phase, gradient-projection iterations approximately minimize a bound-constrained augmented Lagrangian function and provide an estimate of the optimal active set. In the second phase, an equality-constrained QP defined by the current active set is approximately minimized in order to generate a second-order search direction. A filter determines the required accuracy of the subproblem solutions and provides an acceptance criterion for the search directions. The resulting algorithm is globally and finitely convergent. The algorithm is suitable for large-scale problems with many degrees of freedom, and provides an alternative to interior-point methods when iterative methods must be used to solve the underlying linear systems. Numerical experiments on a subset of the CUTEr QP test problems demonstrate the effectiveness of the approach.

Key words. large-scale optimization, quadratic programming, gradient projection, active-set methods, filter methods, augmented Lagrangian

AMS subject classifications. 65K05, 90C06, 90C20, 90C26, 90C52

DOI. 10.1137/060669930

1. Introduction.

Quadratic programs (QPs) play a fundamental role in optimization. They are useful across a rich class of applications, such as the simulation of rigid multibody dynamics [2, 50], optimal control [7, 32, 53], and financial-portfolio optimization [15, 54]. They also arise as a sequence of subproblems within algorithms for solving more general nonlinear optimization problems. Of particular interest for us are sequential quadratic programming (SQP) methods, which have proved to be a reliable approach for general problems (for a recent survey, see Gould and Toint [47]). Our purpose is to develop a QP algorithm that may be used effectively within an SQP framework for solving large-scale nonlinear problems.

Compared to interior-point methods for QPs, active-set methods are especially effective as subproblem solvers within the SQP framework because they can exploit increasingly good starting points in order to reduce the number of iterations required for convergence. Inertia-controlling active-set strategies (see, e.g., [33, 43]) are robust in practice, but their overall efficiency is limited by the number of active-set changes that can be made at each iteration (typically, a single index changes at each iteration). The combinatorial nature of such an approach severely limits its effectiveness on truly large-scale, nonconvex problems that may have many degrees of freedom. However, the robustness and warm-start capability of active-set approaches motivate us to

*Received by the editors September 15, 2006; accepted for publication (in revised form) August 27, 2007; published electronically April 23, 2008. Preliminary versions appeared as Preprint ANL/MCS-P1370-0906 and UBC Department of Computer Science Technical report TR-2007-16.
http://www.siam.org/journals/sisc/30-4/66993.html

†Department of Computer Science, University of British Columbia, Vancouver V6T 1Z4, BC, Canada (mpf@cs.ubc.ca). The research of this author was supported in part by a Discovery Grant from the Natural Science and Engineering Research Council of Canada.

‡Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439 (leyffer@mcs.anl.gov). The work of this author was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under contract W-31-109-ENG-38.

propose a method that is capable of extremely large changes to the active set at each iteration and yet continues to be finitely convergent.

Interior-point methods are often preferred over active-set approaches because they have proved effective for large problems and because they have strong theoretical convergence properties. For convex QPs, interior methods are convergent in polynomial time [56]. However, the key subproblems within these methods lead to linear systems (known as Karush–Kuhn–Tucker (KKT), or saddle-point, systems) that are inherently ill-conditioned [38, Theorem 4.2]. Implementations based on iterative linear solvers need to overcome this ill-conditioning by appealing to specialized preconditioners; this has led to significant research efforts for developing effective preconditioners [41, 12, 11, 10, 16].

In contrast, the KKT systems that arise in active-set methods do not suffer from the artificial ill-conditioning inherent in the barrier term of interior methods. We recognize that preconditioning KKT systems is still an active and open research area, but our expectation is that KKT systems arising in active-set methods will more easily lend themselves to effective preconditioning than those arising in interior-point methods. We are particularly interested in developing methods that have a strong potential to be effective within a matrix-free context. Such methods may have applicability, for example, to the large problems that arise in PDE-constrained optimization with inequality constraints.

With this goal in mind, we propose a new algorithm for solving QPs that is motivated by the computational effectiveness of gradient-projection methods (such as those described by [13, Chapter 2] and [21]) for bound-constrained QPs. A simplistic extension of gradient projection to general QPs would lead to a subproblem that is almost as difficult to solve as the original QP: Each projection of the objective gradient onto the feasible set is itself a QP. Instead, we use the augmented Lagrangian function to transform the QP into a bound-constrained problem on which we can perform inexpensive gradient-projection iterations.

Each iteration of our algorithm has two main phases. The first phase applies inexpensive gradient-projection iterations in order to minimize the augmented Lagrangian function subject to the original problem's bound constraints. This phase encourages rapid changes to the active set and provides an estimate of the optimal active set. With that active-set estimate, the second phase then solves an equality-constrained QP (it is this subproblem that gives rise to the KKT system). A filter method [36] is used to dynamically control the accuracy of the bound-constrained solves, thereby eliminating an arbitrary and sometimes troublesome sequence of parameters commonly used in augmented Lagrangian techniques.

We prove global and finite convergence of the algorithm and show that it identifies the optimal active set in a finite number of iterations. Once this active set has been identified, the algorithm may be interpreted as a Newton iteration on the active set. We present preliminary numerical results that demonstrate the effectiveness of this approach.

1.1. The quadratic program. We consider general QPs of the form

$$\text{(GQP)} \qquad \begin{array}{ll} \underset{x\in\mathrm{R}^n}{\text{minimize}} & c^Tx + \frac{1}{2}x^THx \\ \text{subject to} & Ax = b, \quad x \geq 0, \end{array}$$

where b and c are m- and n-vectors, H is an $n \times n$ symmetric (and possibly indefinite) matrix, and A is an $m \times n$ matrix. Typically, $n \gg m$. QPs with more general upper and lower bounds are easily accommodated by our method.

Notation. Unless otherwise indicated, the 2-norm of a vector v is denoted by $\|v\|$. Subscripts on vectors indicate components, so that v_i is the ith component of v, and if $\mathcal{I}$ is an index set, then $v_{\mathcal{I}}$ is a subvector indexed by $\mathcal{I}$. Unless indicated otherwise, superscripts indicate iterates, so that v^k is the kth iterate. With vector arguments, the functions $\min\{\cdot,\cdot\}$ and $\max\{\cdot,\cdot\}$ are defined componentwise.

We define the *augmented Lagrangian* corresponding to (GQP) as

$$\mathcal{L}_\rho(x,y) = c^Tx + \tfrac{1}{2}x^THx - y^T(Ax-b) + \tfrac{1}{2}\rho\|Ax-b\|^2,$$

where x and the m-vector y are independent variables and $\rho > 0$. The usual Lagrangian function is then $\mathcal{L}_0(x,y)$. When y^k and ρ^k are fixed, we often use the shorthand notation $\mathcal{L}^k(x) := \mathcal{L}_{\rho^k}(x,y^k)$. Define the *first-order multiplier estimate* by

$$\widetilde{y}_\rho(x,y) = y - \rho(Ax-b). \tag{1.1}$$

The derivatives of $\mathcal{L}_\rho$ with respect to x may be written as follows:

$$\nabla_x\mathcal{L}_\rho(x,y) = c + Hx - A^T\widetilde{y}_\rho(x,y), \tag{1.2a}$$

$$\nabla^2_{xx}\mathcal{L}_\rho(x,y) = H + \rho A^TA. \tag{1.2b}$$

We assume that (GQP) is feasible and has at least one point (x^*,y^*) that satisfies the first-order KKT conditions.

Definition 1.1 (first-order KKT conditions). *A pair (x^*,y^*) is a first-order KKT point for* (GQP) *if*

$$\min\{x^*, \nabla_x\mathcal{L}_0(x^*,y^*)\} = 0, \tag{1.3a}$$

$$Ax^* = b. \tag{1.3b}$$

The vector of $z^* := \nabla_x\mathcal{L}_0(x^*,y^*)$ is the set of Lagrange multipliers that corresponds to the bounds $x \geq 0$. Our method remains feasible with respect to the simple bounds, and we define the *active* and *inactive* bound constraints at x by the index sets

$$\mathcal{A}(x) = \{j \in 1,\dots,n \mid x_j = 0\} \qquad \text{and} \qquad \mathcal{I}(x) = \{j \in 1,\dots,n \mid x_j > 0\}.$$

The symbol x^* may denote a (primal) solution of (GQP) and may also be used to denote a limit point of the sequence $\{x^k\}$. Let $\mathcal{A}^* := \mathcal{A}(x^*)$ and $\mathcal{I}^* := \mathcal{I}(x^*)$. For $j \in \mathcal{I}^k$, let H_k be the submatrix formed from the jth rows and columns of H. Similarly, let A_k and A_* be the submatrices formed from the columns of A indexed by $\mathcal{I}^k$ and $\mathcal{I}^*$, respectively.

A vital component of our algorithm is the concept of a filter [36], which we use to determine the required subproblem optimality and to test acceptance during the linesearch procedure. The filter is defined by a collection of tuples together with a rule that must be enforced among all entries maintained in the filter. We denote the filter at the kth iteration by $\mathcal{F}^k$; it is fully defined in section 2.1.

1.2. Related work. Our method is related to a number of nonlinear programming approaches. The two-phase aspect of our method is reminiscent of sequential linear programming/equality quadratic programming methods, which have received much attention recently. For examples of such approaches, see Fletcher and Sainz de la Maza [35] and, more recently, Chin and Fletcher [19] and Byrd et al. [17]. A common approach of these methods is to solve a relatively inexpensive linear programming subproblem in order to estimate the optimal active set, and then solve an equality-constrained QP to obtain a search direction in a subspace. The idea of using

Algorithm 1: Outline of QP Filter Method (QPFIL).

initialization: $k \leftarrow 0$, x^0 given, initialize $\mathcal{F}^0$

while *not optimal* **do**

1. Approximately minimize $\mathcal{L}^k(x)$, subject to $x \geq 0$, to find an $\widetilde{x}^k$ acceptable to $\mathcal{F}^k$.
2. Identify an active set $\mathcal{A}^k$ and update the penalty parameter ρ^{k+1}.
3. Update the multiplier estimate: $\widetilde{y}^k \leftarrow y^k - \rho^k(A\widetilde{x}^k - b)$.
4. Solve an equality-constrained QP for a second-order step $(\Delta x, \Delta y)$.
5. Linesearch: find α such that $(\widetilde{x}^k + \alpha\Delta x, \widetilde{y}^k + \alpha\Delta y)$ is acceptable to $\mathcal{F}^k$.
6. Update iterates: $(x^{k+1}, y^{k+1}) \leftarrow (\widetilde{x}^k + \alpha\Delta x, \widetilde{y}^k + \alpha\Delta y)$.
7. Update filter $\mathcal{F}^{k+1}$.
8. $k \leftarrow k + 1$.

gradient projection to predict the optimal active set has been used in the context of bound-constrained QPs (i.e., with no general linear constraints) by Moré and Toraldo [55] and by Friedlander and Martínez [39], among others. Bound-constrained QP solvers have also been considered by [6, 20, 26, 27, 25].

Our algorithm may be interpreted as a second-order version of the classical augmented Lagrangian algorithm for nonlinear programming, as implemented in the software package LANCELOT [22]. We follow [40] and use the term *bound-constrained Lagrangian* (BCL) for these methods because they involve only bound constraints on each subproblem. Typically the original bound constraints are repeated verbatim in each subproblem and enforced at all iterations. See [13, Chapter 2] and [23] for an overview of BCL methods. BCL methods for convex QPs with general constraints have recently been considered by Dostál, Friedlander, and Santos [28, 29, 30] and by Delbos and Gilbert [24]. Active-set methods for solving large-scale nonconvex QPs include Galahad's QPA [48], BQPD [34], and SQOPT [42]; see Gould and Toint [49] for a recent survey.

2. Augmented Lagrangian filter algorithm for QPs. Our algorithm differs from classical BCL methods in three important ways: First, the main role of the augmented Lagrangian minimization in our algorithm is to provide an estimate of the optimal active set, which is used to define an equality-constrained QP that is subsequently solved for a second-order step. The second-order step improves both the reliability and the convergence rate of BCL methods. Second, we use a filter to control various aspects of the algorithm related to global convergence. The filter allows us to dispense with two forcing sequences commonly used in BCL methods (the subproblem tolerance and the accept/reject threshold for updating the Lagrange multipliers). It also provides a nonmonotone globalization strategy that is more likely to accept steps computed by inexact solutions. Third, we exploit the special structure of the QP problem to obtain estimates of the required penalty parameter. These estimates are more adaptive than traditional penalty update schemes, which may overestimate the penalty parameter. Algorithm 1 outlines the main steps of our approach, which we call QPFIL.

At this point, the careful reader may ask why QPFIL uses *both* a filter and a penalty function—after all, filter methods are meant to replace penalty functions (such as the augmented Lagrangian). The reason is simple: We use the augmented Lagrangian to transform the general QP into a bound-constrained QP so that we can use efficient gradient-projection techniques to derive an active-set estimate. We emphasize that global convergence is enforced by the filter and not through the augmented Lagrangian.

A crucial feature of QPFIL is its suitability for high-performance computing. The two computational kernels of the algorithm are the bound-constrained minimization of the augmented Lagrangian function (step 1) and the solution of an equality-constrained QP (step 4). Scalable tools that perform well on high-performance architectures exist for both steps. For example, TAO [9] and PETSc [4, 3] are suitable, respectively, for the bound-constrained subproblem and the equality-constrained QP. In the remainder of this section we give details of each step of the QPFIL algorithm.

2.1. An augmented Lagrangian filter. The iterations of a BCL method for nonconvex optimization typically are controlled by two fundamental forcing sequences that ensure convergence to a solution. A decreasing sequence, $\omega^k \to 0$, determines the required optimality of each subproblem solution and controls the convergence of the dual infeasibility (see (1.3a)). The second decreasing sequence, $\eta^k \to 0$, tracks the primal infeasibility (see (1.3b)) and determines whether the penalty parameter ρ^k should be increased or left unchanged.

In the definition of our filter we use quantities that are analogous to ω^k and η^k:

$$\begin{aligned} \omega(x,y) &:= \|\min\{x, \nabla_x \mathcal{L}_0(x,y)\}\|, \\ \eta(x) &:= \|Ax - b\|, \end{aligned}$$

which are based on the optimality and feasibility of a current pair (x, y). As we prove in section 3, such a choice allows us to dispense with the sequences normally found in BCL methods and instead defines these sequences implicitly. We observe that the filter will generally be less conservative than BCL methods in the acceptance of a current subproblem solution or multiplier update.

Note that $w(x, y)$ is based on the gradient of the Lagrangian function, not on the augmented Lagrangian. Thus, our decision on when to exit the minimization of the current subproblem is based on the optimality of the current subproblem iterate for the original problem, rather than on the optimality of the current subproblem, as is usually the case in BCL methods. This approach ensures that the subproblem iterations (defined below) always generate solutions that are acceptable to the filter. Another advantage of this definition is that the filter is, in effect, independent of the penalty parameter ρ^k and hence does not need to be updated if ρ^k is increased.

In the remainder of the paper we use the abbreviations

$$\omega^k := \omega(x^k, y^k) \quad \text{and} \quad \eta^k := \eta(x^k).$$

DEFINITION 2.1 (augmented Lagrangian filter). *The following rules define an augmented Lagrangian filter:*

1. *A pair (ω', η') dominates another pair (ω, η) if $\omega' \le \omega$ and $\eta' \le \eta$, and at least one inequality holds strictly.*
2. *A filter $\mathcal{F}$ is a list of pairs (ω, η) such that no pair dominates another.*
3. *A filter $\mathcal{F}$ contains an entry (called the upper bound)*

$$(\bar\omega, \bar\eta) = (U, 0), \tag{2.1}$$

   where U is a positive constant.
4. *A pair (x', y') is* acceptable to the filter $\mathcal{F}$ *if and only if*

$$\omega' \le \beta\omega^\ell \quad \text{or} \quad \eta' \le \beta\eta^\ell - \gamma\omega' \tag{2.2}$$

   for each $(\omega^\ell, \eta^\ell) \in \mathcal{F}$, where $\beta, \gamma \in (0, 1)$ are constants.

We use the shorthand notation $\ell \in \mathcal{F}$ to imply that $(\omega^\ell, \eta^\ell) \in \mathcal{F}$.

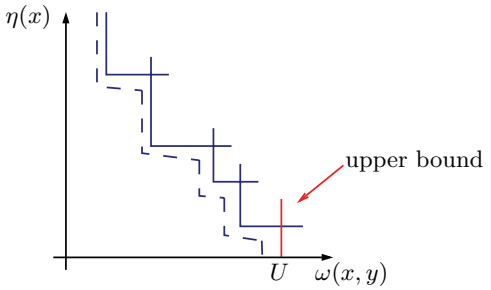


FIG. 2.1. *A typical filter. All pairs (ω, η) that are below and to the left of the envelope (dashed line) are acceptable to the filter (cf.* (2.2)*).*

A typical filter is illustrated in Figure 2.1. Typical values for the envelope constants are $\beta = 0.999$, $\gamma = 0.001$. A suitable choice for the upper bound U in (2.1) is $U = \delta \max\{1, \omega^0\}$, with $\delta = 1.25$. Filter methods are typically insensitive to the choice of these parameters, and most importantly, these parameters are not problem-dependent, unlike penalty parameters which must be chosen with more care. We note that (2.2) creates a sloping envelope around the filter. Together with (2.1), this implies that a sequence $\{(\omega^k, \eta^k)\}$ of pairs each acceptable to $\mathcal{F}^k$ must satisfy $\omega^k \to \omega^* = 0$. If the second condition in (2.2) were weakened to $\eta^{k+1} \leq \beta\eta^\ell$, then the sequence of pairs acceptable to $\mathcal{F}^k$ could accumulate to points where $\eta^k \to \eta^* = 0$, but which are nonstationary because $\omega^k \to \omega^* > 0$.

A consequence of $\eta(x) \geq 0$ and the sloping envelope is that the upper bound $(U, 0)$ is theoretically unnecessary—the sloping envelope implies an upper bound $U = \eta_{\min}/\gamma$, where $\eta_{\min}$ is the least η^ℓ for all $\ell \in \mathcal{F}$. In practice, however, we impose the upper bound U in order to avoid generating entries with excessively large values ω^k.

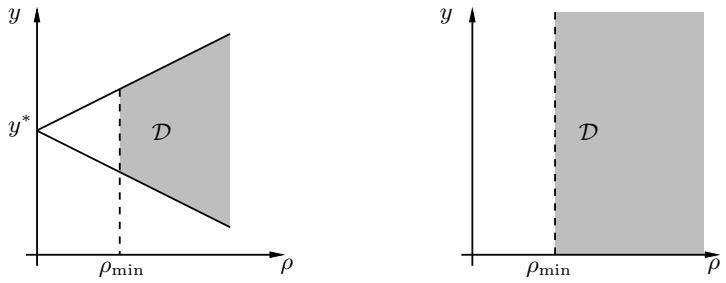
We remark that the axes in the augmented Lagrangian filter appear to be the reverse of the usual definition: Feasibility is on the vertical axis instead of the horizontal axis, as it typically appears in the literature. This reflects the dual view of the augmented Lagrangian: It can be shown that $Ax - b$ is a steepest descent direction at x for the augmented Lagrangian [14, section 2.2], and that $\omega(x, y)$ is the dual feasibility error. This definition of the filter is similar to the one used in [45]. The gradient of the Lagrangian has also been used in the filter by M. Ulbrich, S. Ulbrich, and Vicente [59], together with a centrality measure, in the context of interior-point methods.

2.2. Active-set prediction and second-order steps. Let $\widetilde{x}^k$ be an approximate minimizer of the augmented Lagrangian $\mathcal{L}^k$ at iteration k. We use this solution to derive an active-set estimate $\mathcal{A}^k := \mathcal{A}(\widetilde{x}^k)$, which in turn is used to define an equality-constrained QP (EQP) in the free variables, which are indexed by $\mathcal{I}^k := \mathcal{I}(\widetilde{x}^k)$. The variables $\mathcal{A}^k$ are held fixed at the active bounds.

A second-order correction to $\widetilde{x}^k$ in the space of free variables may be found by solving the following EQP for $\Delta x = (\Delta x_{\mathcal{A}}, \Delta x_{\mathcal{I}})$:

$$(\mathrm{EQP}_k) \qquad \begin{array}{ll} \underset{\Delta x}{\text{minimize}} & c^T(\widetilde{x}^k + \Delta x) + \frac{1}{2}(\widetilde{x}^k + \Delta x)^T H(\widetilde{x}^k + \Delta x) \\ \text{subject to} & A(\widetilde{x}^k + \Delta x) = b, \quad \Delta x_{\mathcal{A}^k} = 0. \end{array}$$

Equivalently, a second-order search direction from the current point $(\widetilde{x}^k, \widetilde{y}^k)$ is gener-



(a) Nonlinear constraints (b) Linear constraints

FIG. 2.2. *The sets $\mathcal{D}$ illustrate the required penalty parameter for the BCL method when the constraints are either nonlinear or linear.*

ated from the (first-order) optimality conditions of (EQP_k):

$$\begin{pmatrix} -H_k & A_k^T \\ A_k & \end{pmatrix} \begin{pmatrix} \Delta x_{\mathcal{I}} \\ \Delta y \end{pmatrix} = \begin{pmatrix} [c + H\widetilde{x}^k]_{\mathcal{I}^k} - A_k^T \widetilde{y}^k \\ b - A\widetilde{x}^k \end{pmatrix}. \tag{2.3}$$

A projected search in the full space is then based on the vector $(\Delta x, \Delta y)$.

Note that step 1 of Algorithm 1 requires that the approximate augmented Lagrangian minimizer $\widetilde{x}^k$ be acceptable to the filter. Moreover, as we demonstrate in section 3.2, the first-order multiplier estimate $\widetilde{y}^k$ must also be acceptable to the filter. These two properties ensure that even if a linesearch along $(\Delta x, \Delta y)$ fails to obtain a positive steplength α such that $(\widetilde{x}^k + \alpha\Delta x, \widetilde{y}^k + \alpha\Delta y)$ is acceptable to the filter, the algorithm can still make progress with the first-order step alone. In this case, $\alpha = 0$, and the algorithm relies on the progress of the standard BCL iterations.

2.3. Estimating the penalty parameter. It is well known that BCL methods, under standard assumptions, converge for all large enough values of the penalty parameter ρ^k. The threshold value $\rho_{\min}$ is never computed explicitly; instead, BCL methods attempt to discover the threshold value by increasing ρ^k in stages. Typically the norm of the constraint violation is used to guide the decisions regarding when to increase the penalty parameter: A linear decrease (as anticipated by the BCL local convergence theory) signals that the penalty parameter may be held constant; less than linear convergence—or a large increase in constraint violations—indicates that a larger ρ^k is needed.

When the constraints are nonlinear, the penalty-parameter threshold and the initial Lagrange multiplier estimates are closely coupled. Poor estimates y^k of y^* imply that a larger ρ^k is needed to induce convergence. This coupling is fully described by Bertsekas [13, Proposition 2.4]. When the constraints are linear, however, the Lagrange multipliers do not appear in (1.2b), and we see that y^k and ρ^k are essentially decoupled—the curvature of $\mathcal{L}^k$ can be influenced by changing ρ^k alone. This observation is illustrated in Figure 2.2, in which the left figure corresponds to nonlinear constraints and the right figure to linear constraints. The regions in the penalty/multiplier plane for which BCL methods converge are indicated by the shaded regions $\mathcal{D}$. The result below provides an explicit threshold value $\rho_{\min}$ needed to ensure that the Hessian of the augmented Lagrangian is positive definite (a positive multiple of $\rho_{\min}$ is enough to induce convergence). Let $\lambda_{\min}(\cdot)$ and $\sigma_{\min}(\cdot)$, respectively, denote the leftmost eigenvalue and the smallest singular value of a matrix.

LEMMA 2.2. *Suppose that $p^THp > 0$ for all nonzero p such that $Ap = 0$ and A has full row rank. Then $H + \rho A^TA$ is positive definite if and only if*

$$\rho > \rho_{\min} := \lambda_{\min}\big(A\big(H + \gamma A^TA\big)^{-1}A^T\big)^{-1} - \gamma I \tag{2.4}$$

for any $\gamma \geq 0$ such that $H + \gamma A^TA$ is nonsingular.

Proof. The required result follows from Bertsekas [13, Proposition 2.5], where the Jacobian and Hessian are taken as constant. ∎

The bound provided by Lemma 2.2 is sharp: It is both necessary and sufficient. However, the formula on the right-hand side of (2.4) is unsuitable for large-scale computation. The following lemma develops an upper bound for the required ρ that is more easily computed.

LEMMA 2.3. *Under the conditions of Lemma* 2.2*,*

$$\rho_{\min} < \frac{\max\{0, -\lambda_{\min}(H)\}}{\sigma_{\min}(A)^2}. \tag{2.5}$$

Proof. Consider unit-norm vectors p such that $Ap \neq 0$. Otherwise, $p^T(H + \rho A^TA)p = p^THp > 0$ for all p such that $Ap = 0$. Let $U = [u_1 \cdots u_m]$ be the orthogonal left-singular vectors of A, and let $\Sigma = \operatorname{diag}(\sigma_i)$ be the singular values, with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \equiv \sigma_{\min} > 0$ (we assume that A has full rank). Then p can be expressed as $p = \sum_{i=1}^m \alpha_i u_i$ with $\sum_{i=1}^m \alpha_i^2 = 1$, for some scalars α_i not all zero. Thus $A^TA = U\Sigma^T\Sigma U^T$, and

$$p^TA^TAp = \left(\sum_{i=1}^m \alpha_i u_i^T\right)(U\Sigma^T\Sigma U^T)\left(\sum_{i=1}^m \alpha_i u_i\right) = \sum_{i=1}^m \alpha_i^2\sigma_i^2. \tag{2.6}$$

Similarly, let $Q = [q_1 \cdots q_n]$ be the orthogonal eigenvectors of H, and let $\Lambda = \operatorname{diag}(\lambda_i)$ be the eigenvalues, with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \equiv \lambda_{\min}$. Then there exist scalars β_i not all zero such that $p = \sum_{i=1}^n \beta_i q_i$ with $\sum_{i=1}^n \beta_i^2 = 1$, and

$$p^THp = \left(\sum_{i=1}^n \beta_i q_i^T\right) H \left(\sum_{i=1}^n \beta_i q_i\right) = \sum_{i=1}^n \beta_i^2\lambda_i. \tag{2.7}$$

Therefore, (2.6) and (2.7) imply that

$$\begin{aligned} p^T(H + \rho A^TA)p &= \sum_{i=1}^n \beta_i^2\lambda_i + \rho\sum_{i=1}^m \alpha_i^2\sigma_i^2 \\ &> \min(0, \lambda_{\min})\sum_{i=1}^n \beta_i^2 + \rho\sigma_{\min}^2\sum_{i=1}^m \alpha_i^2 \\ &= \min(0, \lambda_{\min}) + \rho\sigma_{\min}^2, \end{aligned}$$

and so $H + \rho A^TA$ is positive definite if $\min(0, \lambda_{\min}) + \rho\sigma_{\min}^2 > 0$ or, equivalently, if

$$\rho > \frac{\max(0, -\lambda_{\min})}{\sigma_{\min}^2}. \tag{2.8}$$

Because the bound $\rho_{\min}$ in Lemma 2.2 is sharp, (2.8) implies that (2.5) holds. ∎

For a given active set $\mathcal{A}^k$, Lemma 2.3 implies that ρ^k larger than

$$\rho_{\min}(\mathcal{A}^k) := \frac{\max\{0, -\lambda_{\min}(H_k)\}}{\sigma_{\min}(A_k)^2} \tag{2.9}$$

is sufficient at iteration k to ensure that $\mathcal{L}^k$ is convex in that subspace. Note that this lower bound tends to infinity as the smallest singular value of A_k tends to zero. This property is consistent with (2.4), where we see that if A_k is rank deficient, then the required bound in Lemma 2.2 does not exist. In section 3 we show that for a given optimal active set, a multiple of this bound is required to induce convergence to an optimal solution in our method.

We are not entirely satisfied with (2.9) because it requires an estimate (or at least a lower bound) of the smallest singular value of the current A_k, which can be relatively expensive to compute. One possibility for estimating this value is to use a Lanczos bidiagonalization procedure, as implemented in PROPACK [51].

Ideally, we would compute the penalty value according to (2.4) or (2.5). However, for the size of problems of interest, this approach would be prohibitive in terms of computational effort. In our numerical experiments we have instead used the quantity

$$\rho_{\min}(\mathcal{A}^k) = \max\left\{1,\ \frac{\|H_k\|_1}{\max\left\{\frac{1}{\sqrt{|\mathcal{I}^k|}}\|A_k\|_\infty\ ,\ \frac{1}{\sqrt{m}}\|A_k\|_1\right\}}\right\},$$

where $|\mathcal{I}^k|$ is the number of free variables and m is the number of general equality constraints, as a simple approximation to (2.9). We note that the penalty parameter appears only within the subproblem minimization (step 1 of Algorithm 1), and not in the definition of the filter. If only a rough approximation to (2.9) is available, then a multiple of the approximation might be used so as to increase the likelihood that a large enough quantity is obtained. In the remainder of the paper, we assume that $\rho_{\min}(\mathcal{A}^k)$ is given by (2.9).

2.4. Minimizing the augmented Lagrangian subproblem. Like classical BCL methods, our method generates a sequence of approximate minimizers of the bound-constrained subproblem

$$\underset{x}{\text{minimize}} \quad \mathcal{L}^k(x) \quad \text{subject to} \quad x \geq 0. \tag{2.10}$$

Instead of optimizing the subproblem to a prescribed tolerance, however, each iteration of the inner algorithm approximately optimizes it in stages (i.e., a few iterations of some minimization procedure are applied), so that at each iteration j of the inner algorithm, the current iterate x^j satisfies the approximate optimality conditions

$$\|\min\{x, \nabla_x \mathcal{L}_{\rho^j}(x, \bar{y})\}\|_\infty \leq \epsilon^j, \tag{2.11}$$

where $\bar{y}$ is the latest multiplier estimate y^k. The only requirement for the sequence of approximate minimizations is that they eventually solve the subproblem in the limit, and thus that $\epsilon^j \to 0$. The iterate x^j and the implied first-order multiplier (1.1) are tested for acceptability against the current filter. The inner-minimization algorithm is described in Algorithm 2.

The penalty parameter ρ^j is checked at each inner iteration to ensure that it satisfies the bound implied by Lemma 2.2 (see steps 5 and 7). If the current submatrix

Algorithm 2: Bound-Constrained Lagrangian Filter (BCLFIL).

Inputs: x^0, $\bar{y}$, ρ^1, $\mathcal{F}$ **Outputs:** $\widetilde{x}$, $\widetilde{y}$, $\widetilde{\rho}$

Set $\alpha \in [0, 1)$, $j \leftarrow 0$

repeat

1 $j \leftarrow j + 1$

2 Choose $\epsilon^j > 0$ such that $\lim_{j\to\infty} \epsilon^j = 0$

3 Find a point x^j that satisfies (2.11) [approximately solve (2.10)]

4 $\mathcal{A}^j \leftarrow \mathcal{A}(x^j)$ [update active set]

5 **if** $\sigma_{\min}(A_j) = 0$ **then**

6 Find a point x^j that satisfies (2.13) [feasibility restoration]

7 **else if** $\rho^j < 2\rho_{\min}(\mathcal{A}^j)$ **then**

8 $\rho^{j+1} \leftarrow 2\rho_{\min}(\mathcal{A}^j)$ [increase penalty parameter]

9 **else**

10 $y^j \leftarrow \bar{y} - \rho^j(Ax^j - b)$ [provisional multiplier update]

11 $(\omega^j, \eta^j) \leftarrow \big(\omega(x^j, y^j), \eta(x^j)\big)$ [update primal-dual infeasibility]

12 **if** (ω^j, η^j) is acceptable to $\mathcal{F}$ **then**
return $\widetilde{x} \leftarrow x^j$, $\widetilde{y} \leftarrow y^j$, $\widetilde{\rho} \leftarrow \rho^j$

13 $\rho^{j+1} \leftarrow \rho^j$ [keep penalty parameter]

until converged

A_j is rank deficient (i.e., $\sigma_{\min}(A_j) = 0$), then there does not exist a finite ρ that makes the reduced Hessian positive definite. In that case, we are not assured that reducing the augmented Lagrangian brings the next iterate any closer to optimality of the original subproblem. Instead, we make progress towards feasibility of the iterates by approximately solving the minimum infeasibility problem

$$\underset{x}{\text{minimize}} \quad \tfrac{1}{2}\|Ax - b\|^2 \quad \text{subject to} \quad x \geq 0, \tag{2.12}$$

and we thus require that x^j satisfy the approximate necessary and sufficient condition

$$\|\min\{x, A^T(Ax - b)\}\|_\infty \leq \epsilon^j. \tag{2.13}$$

The point $x^j \geq 0$ solves the minimum infeasibility problem if $A_j^T(A_j x^j_{\mathcal{I}^j} - b) = 0$, which can be satisfied at infeasible points if A_j is rank deficient.

An alternative to step 6 of Algorithm 2 is to increase ρ^j by a fixed multiple. A similar strategy is used in the method suggested in [30], where ρ^j is increased if the current iterate is not "extended regular." With this update, it can be shown that if $\rho^j \to \infty$, then every limit point x^* of x^j is either a KKT point of (GQP) or a solution of (2.12) (see Theorem A.1). The analysis given in [30] shows that x^* continues to be a solution of the original QP, but this conclusion depends crucially on the strict convexity of (GQP)—an assumption that we do not make here.

In classical BCL methods, the gradient of the augmented Lagrangian at the latest iterate x^j and the latest multiplier estimate $\bar{y}$ is used to test termination of the inner iterations. The test in step 12 of Algorithm 2 is based on the norm of the (usual) Lagrangian function at x^j, but it differs from BCL in using the first-order multiplier estimate $y^j = \bar{y} - \rho^j(Ax^j - b)$. We note that the identity $\nabla_x \mathcal{L}_{\rho^j}(x^j, \bar{y}) = \nabla_x \mathcal{L}_0(x^j, y^j)$ implies that the quantities used to test termination in Algorithm 2 and in classical BCL methods are in fact identical. Algorithm 2 additionally uses the current primal infeasibility η^j as a criterion. The inner minimization terminates when the current

Algorithm 3: QP Filter Method (QPFIL).

Inputs: x^0, y^0 **Outputs:** x^*, y^*

Set penalty parameter $\rho^0 > 0$ and positive filter envelope parameters $\beta, \gamma < 1$.
Set filter upper bound $U \leftarrow \gamma \max\{1, \|Ax^0 - b\|\}$, and add $(U, 0)$ to filter $\mathcal{F}^0$.
Set minimum steplength $\alpha_{\min} > 0$.
Compute infeasibilities $\omega^0 \leftarrow \omega(x^0, y^0)$ and $\eta^0 \leftarrow \eta(x^0)$.
$k \leftarrow 0$
1 **if** $\omega^0 > 0$ and $\eta^0 > 0$ **then** add (ω^0, η^0) to $\mathcal{F}^0$.
while *not optimal* **do**
2 $k \leftarrow k + 1$
3 $(\widetilde{x}^k, \widetilde{y}^k, \rho^k) \leftarrow$ **BCLFIL**$(x^{k-1}, y^{k-1}, \rho^{k-1}, \mathcal{F}^{k-1})$
4 $\mathcal{A}^k \leftarrow \mathcal{A}(\widetilde{x}^k)$
5 Find $(\Delta x^k, \Delta y^k)$ that solves (2.3)
6 Find $\alpha^k \in [\alpha_{\min}, 1]$ such that $(\widetilde{x}^k + \alpha \Delta x^k, \widetilde{y}^k + \alpha \Delta y^k)$ is acceptable to $\mathcal{F}^k$
7 **if** linesearch failed **then**
8 $(x^k, y^k) \leftarrow (\widetilde{x}^k, \widetilde{y}^k)$ [keep first-order iterates]
else
9 $(x^k, y^k) \leftarrow (\widetilde{x}^k + \alpha^k \Delta x^k, \widetilde{y}^k + \alpha \Delta y^k)$ [second-order update]
10 $(\omega^k, \eta^k) \leftarrow \big(\omega(x^k, y^k), \eta(x^k)\big)$ [compute infeasibilities]
11 **if** $\omega^k > 0$ **then**
12 $\mathcal{F}^k \leftarrow \mathcal{F}^{k-1} \cup \{(\omega^k, \eta^k)\}$
13 Remove redundant entries from $\mathcal{F}^k$
14 **if** $\eta^k = 0$ **then** update upper bound U
return $x^* \leftarrow x^k$, $y^* \leftarrow y^k$

iterates are acceptable to the filter and the penalty parameter is large enough for the current active set.

To establish that our algorithm finitely identifies the optimal active set (see section 4), we assume that each approximate minimization reduces the objective by at least as much as does a Cauchy point of a projected-gradient method (see, e.g., [57, section 16.6]). This is a mild assumption that is satisfied by most globally convergent bound-constrained solvers. In practice, we perform one or two steps of a bound-constrained optimization algorithm and then test the acceptability of (ω^j, η^j) to the filter. This requirement is often weaker than traditional augmented Lagrangian methods, which at each outer iteration must reduce the projected gradient beyond a specified tolerance that goes to zero; in contrast, here the inner-iteration tolerances are independent across outer iterations.

2.5. Detailed algorithm statement. The proposed algorithm is structured around *outer* and *inner* iterations. The outer iterations handle management of the filter, the solution of (EQP$_k$), and the subsequent linesearch. The inner iterations minimize the augmented Lagrangian function, update the multipliers and the penalty parameter, and identify a candidate set of active constraints used to define (EQP$_k$) for the outer iteration. Thus, each inner iteration performs steps 1–3 of Algorithm 1.

In step 6 of Algorithm 3 we perform a filter linesearch by trying a sequence of steps $\alpha = \gamma^i$, $i = 0, 1, 2, \ldots$, for some constant $\gamma \in (0, 1)$ until an acceptable point is found or until $\alpha < \alpha_{\min}$, where $\alpha_{\min} > 0$ is a constant parameter. The parameter $\alpha_{\min}$ is needed because the first-order point $(\widetilde{x}^k, \widetilde{y}^k)$ could lie in a corner of the filter

with the second-order step implying a step into the filter. In that case there exists no $\alpha > 0$ that yields an acceptable step. Other ways of deciding when to terminate the linesearch are possible, based, for example, on requiring that the new filter area induced by the linesearch step be larger than the new filter area induced by the first-order step.

The filter update in step 12 of Algorithm 3 removes redundant entries that are dominated by a new entry. The upper bound $(U, 0)$ also allows us to manage the number of filter entries that we wish to store. If this number is exceeded, then we can reset the upper bound as $U = \max_\ell\{\omega^\ell \mid \omega^\ell \in \mathcal{F}^k\}$ and subsequently delete dominated entries from $\mathcal{F}^k$, thus reducing the number of filter entries.

3. Global convergence. Global convergence of Algorithm 3 (QPFIL) is based on progress made by the inner iterations of step 3. The second-order updates in steps 5–9 serve only to accelerate convergence. Therefore, we can establish global convergence of QPFIL by analyzing a first-order version of the algorithm that does not use the second-order updates. The following assumption holds implicitly throughout.

ASSUMPTION 3.1. *The sequences $\{x^j\}$ and $\{x^k\}$ generated by Algorithms 2 and 3 lie in a compact set. Hence, each sequence has at least one limit point.*

3.1. Preliminaries. When A_* has full rank, define the *least-squares multiplier estimate* $\widehat{y}(x)$ as the unique solution of the least-squares problem

$$\widehat{y}(x) := \arg\min_y \|[c + Hx]_{\mathcal{I}^*} - A_*^T y\|. \tag{3.1}$$

Because the least-squares multiplier estimate is unique, there exists a positive constant α_1 such that

$$\|\widehat{y}(x) - \widehat{y}(x^*)\| \le \alpha_1 \|x - x^*\|. \tag{3.2}$$

Note that the definition of $\widehat{y}$ requires a priori knowledge of the bounds that are active at the solution; however, $\widehat{y}$ is used only for analysis and is never computed.

LEMMA 3.2. *Suppose that A_* has full rank and y is an approximate least-squares solution of* (3.1) *for some x. Then there exists a positive constant α_2 such that*

$$\|\widehat{y}(x) - y\| \le \alpha_2 \|[c + Hx]_{\mathcal{I}^*} - A_*^T y\|. \tag{3.3}$$

Proof. Let $\widehat{r}(x)$ and r be the least-squares residuals associated with $\widehat{y}(x)$ and y, respectively, so that $A_*^T\widehat{y}(x) + \widehat{r}(x) = [c + Hx]_{\mathcal{I}^*}$ and $A_*^T y + r = [c + Hx]_{\mathcal{I}^*}$. Then

$$A_*^T(\widehat{y}(x) - y) + \widehat{r}(x) - r = 0,$$

and because $A_*\widehat{r}(x) = 0$, it follows that $A_* A_*^T(\widehat{y}(x) - y) + A_* r = 0$. Because A_* has full rank, it is straightforward to show that there exists a positive constant α_2 such that $\|\widehat{y}(x) - y\| \le \alpha_2 \|r\|$, and the required result follows immediately from the definition of r. ∎

The next result shows how a sequence of multiplier estimates is related to the least-squares multiplier estimates.

LEMMA 3.3. *Let $\{\omega^k\}$ and $\{\rho^k\}$ be sequences of positive scalars where $\omega^k \to 0$. Let $\{x^k\}$ and $\{y^k\}$ be any sequences of n-vectors and m-vectors, respectively, that together satisfy*

$$\|\min\{x^k, \nabla_x \mathcal{L}_0(x^k, y^k)\}\|_\infty \le \omega^k. \tag{3.4}$$

Let x^* be any limit point of $\{x^k\}$ with an associated sequence of indices $\mathcal{K}$. Suppose that A_* has full rank and let $y^* := \widehat{y}(x^*)$. Then there are positive constants α_1 and α_2 such that

$$\|y^k - y^*\| \le \beta^k := \alpha_1 \|x^k - x^*\| + \alpha_2 \omega^k \tag{3.5}$$

for all $k \in \mathcal{K}$ large enough.

Proof. Set $z^k := \nabla_x \mathcal{L}_0(x^k, y^k)$. For $k \in \mathcal{K}$ large enough, x^k is sufficiently close to x^* that $x_i^k > 0$ if $x_i^* > 0$. Then for such k, (3.4) and $\omega^k \to 0$ imply that $\min\{x_i^k, z_i^k\} = z_i^*$, so that

$$\|z^k_{\mathcal{I}^*}\| \le \|\min\{x^k, z^k\}\| \le \sqrt{n}\ \omega^k. \tag{3.6}$$

We now derive (3.5). From the triangle inequality,

$$\|y^k - y^*\| \le \|\widehat{y}(x^k) - y^k\| + \|\widehat{y}(x^k) - y^*\|. \tag{3.7}$$

Also, (3.3) (with $x = x^k$ and $y = y^k$) and (3.6) together imply that

$$\|\widehat{y}(x^k) - y^k\| \le \alpha_2 \omega^k.$$

Substituting this and (3.2) (with $x = x^k$) into (3.7), we obtain (3.5). ∎

3.2. Convergence of inner iterations. We expect that the usual behavior of Algorithm 2 will be to terminate finitely. However, as the next theorem proves, if the algorithm does not terminate, then the inner iterations converge to a KKT point of (GQP) or they converge to a solution of the minimum infeasibility problem (2.12). For this section only, let

$$y^j := \bar{y} - \rho^j (Ax^j - b) \quad \text{and} \quad z^j := \nabla_x \mathcal{L}_0(x^j, y^j).$$

THEOREM 3.4 (convergence of inner iterations). *Let $\{x^j\}$ and ρ^k be sequences generated by Algorithm* 2. *Then the algorithm terminates finitely, or every limit point x^* of $\{x^j\}$ is a KKT point of* (GQP) *or solves* (2.12).

Proof. We first consider the case where step 5 tests true finitely many times and then treat separately the two other cases where step 5 tests true infinitely many times (and hence $\rho^j \to \infty$) and A_* is either full rank or not.

Case 1 (step 5 tests true finitely many times). In this case, the alternative steps 7 or 9 must evaluate true for all j large enough. But there are only finitely many different active sets, and so step 5 can evaluate true only finitely many times. Hence, $\{\rho^j\}$ remains bounded and step 12 is tested for all j large enough. Consider only such j. Because each x^j satisfies (2.11), steps 10 and 11 ensure that $\omega^j \to 0$. Moreover, for every $\ell \in \mathcal{F}$, $\omega^\ell > 0$ (see steps 1 and 11 of Algorithm 3), and so ω^j must be acceptable to the filter (see (2.2)) after finitely many iterations. Hence, Algorithm 2 exits finitely.

Case 2 (step 5 tests true infinitely many times). In this case, each x^j in some subsequence $\mathcal{J}$ satisfies (2.13). Because $\epsilon^j \to 0$, the limit point x^* associated with the sequence $\mathcal{J}$ satisfies (2.13), and it must therefore be a solution of (2.12). ∎

In Theorem A.1 (see the appendix) we give an analogous convergence proof for a slightly modified version of Algorithm 2 that offers an alternative to steps 5–6. See [1] for a related convergence analysis that relies on a set of different assumptions.

The hypotheses of Theorem 3.4 can fail to hold if there are no convergent subsequences (i.e., Assumption 3.1 fails to hold) or if Algorithm 2 breaks down because no iterate x^j can be found to satisfy the stopping condition (2.11). For example, the subproblem is unbounded below, which can happen if there exists a nonzero and nonnegative vector d such that $d^T H d < 0$ and $Ad = 0$.

3.3. Convergence of first-order algorithm. For this section only, we consider a simplified algorithm that skips the second-order update (steps 5–10 of Algorithm 3). In this case, $(x^k, y^k) \equiv (\widetilde{x}^k, \widetilde{y}^k)$, and we refer to the sequence $\{(x^k, y^k)\}$ of augmented Lagrangian minimizers and multiplier estimates as *Cauchy points*; our intent is to emphasize that these solutions can be interpreted as steepest-ascent steps of the augmented Lagrangian function and thus can yield only linear convergence.

We prove that the first-order sequence $\{(x^k, y^k)\}$ generated in step 3 converges to a stationary point of (GQP). This result is of interest also within the context of more established BCL methods because it illustrates how a filter can be used in place of the two arbitrary forcing sequences (ω^k and η^k) commonly associated with augmented Lagrangian methods.

We show that the sequence of penalty parameters ρ^k is bounded, and that every limit point of the primal-dual pair (x^k, y^k) satisfies (1.3a) and is thus dual feasible.

LEMMA 3.5. *The penalty parameter is updated finitely often.*

Proof. This follows from the fact that there exist only a finite number of different active sets that could result in a penalty-parameter update. ∎

LEMMA 3.6. *Any limit point* (x^*, y^*) *of* $\{(x^k, y^k)\}$ *satisfies* $\omega^k \equiv \omega(x^k, y^k) \to 0$.

Proof. We consider two mutually exclusive cases, depending on whether a finite or an infinite number of entries are added to the filter. If a finite number of entries are added to the filter (i.e., if step 11 of Algorithm 3 tests true only finitely many times), then it follows that $\omega^k = 0$ for all k sufficiently large. The required result then follows immediately. If, on the other hand, an infinite number of entries (ω^k, η^k) are added to the filter, then the required result follows from [19, Lemma 1]—where we take $f(x) = \eta(x)$ and $h(x) = \omega(x, y)$—because $\eta(x)$ is trivially bounded below. ∎

The following theorem is our main convergence result on the sequence of Cauchy points.

THEOREM 3.7 (global convergence with single limit point). *Consider a version of Algorithm* 3 *that skips steps* 5–10. *Assume that the algorithm generates a sequence of Cauchy points* $\{(x^k, y^k)\}$, *and that* x^* *is the* single *limit point of* $\{x^k\}$. *Then* $y^k \to y^*$, *where* $y^* := \widehat{y}(x^*)$, *and* (x^*, y^*) *is a KKT point of (GQP).*

Proof. Step 3 of Algorithm 3, together with Lemma 3.6, ensures that each x^k, y^k, ρ^k, and ω^k, for $k \in \mathcal{K}$, satisfy the conditions of Lemma 3.3. Then (3.4)–(3.5) hold, and $y^k \to y^*$, as required. Because $\omega^k \to 0$, (3.4) implies that

$$0 \leq \lim_{k\to\infty} \|\min\{x^k, \nabla_x \mathcal{L}_0(x^k, y^k)\}\|_\infty \leq \lim_{k\to\infty} \omega^k = 0,$$

and so $\min\{x^*, \nabla_x \mathcal{L}_0(x^*, y^*)\} = 0$. Therefore, (x^*, y^*) satisfies (1.3a).

The "single-limit-point" assumption on x^k and the definition of $y^k = y^{k-1} - \rho^k(Ax^k - b)$ (see step 10 of Algorithm 2) imply that $\|y^k - y^{k-1}\| \to 0$. By Lemma 3.5, ρ^k is bounded for all k, so

$$\|y^k - y^{k-1}\| = \rho^k \|Ax^k - b\| \to 0, \tag{3.8}$$

and x^* satisfies (1.3b). Hence, (x^*, y^*) is a KKT pair of (GQP). ∎

Recall that (GQP) is nonconvex, and hence the subproblem may have many stationary points. The single-limit-point assumption of Theorem 3.7 excludes the situation in which consecutive minimizations of the augmented Lagrangian subproblem converge to different stationary points. Otherwise, the corresponding Lagrange multiplier updates would not have a limit point and (3.8) would not hold. We could relax the single-limit-point assumption if we instead assume that the subproblem

solver finds a stationary point closest in norm to x^k. Such a requirement cannot be verified in practice, but depending on the subproblem solver, it is, arguably, often satisfied. A similar assumption is made implicitly in a classical proof of convergence of the augmented Lagrangian method: Bertsekas [13, Proposition 2.4] assumes that all minimizers of the augmented Lagrangian fall within a small neighborhood. The single-limit-point assumption made in Theorem 3.7 considerably simplifies the analysis and leads to similar conclusions. In the case where (GQP) is convex, every subproblem has a unique minimizer and neither assumption is required.

If we instead assume that a second-order sufficiency condition exists at a limit point x^*, we can drop the single-limit-point assumption. In effect, the following theorem shows that second-order points are attractors, and the algorithm generates increasingly better Lagrange multiplier estimates.

THEOREM 3.8 (global convergence with second-order sufficiency). *Consider a version of Algorithm* 3 *that skips steps* 5–10. *Assume that the algorithm generates a sequence of Cauchy points* $\{(x^k, y^k)\}$ *and that a limit point* x^* *satisfies the second-order sufficiency condition*

$$p^T H p > 0 \text{ for all } p \neq 0 \text{ satisfying } Ap = 0 \text{ with } p_j = 0 \text{ for all } j \in \mathcal{A}^*. \tag{3.9}$$

Then there exist positive constants $\delta_1, \delta_2, \delta_3$, *and a positive constant* $\gamma < 1$ *such that*

$$\|y^k - y^*\| \leq \delta_1 \omega^k + \gamma \|y^{k-1} - y^*\|, \tag{3.10a}$$

$$\|x^k - x^*\| \leq \delta_2 \omega^k + \delta_3 \|y^{k-1} - y^*\|, \tag{3.10b}$$

$$\rho^k \|Ax^k - b\| \leq \delta_1 \omega^k + (\gamma + 1)\|y^{k-1} - y^*\|, \tag{3.10c}$$

where (x^*, y^*) *is a KKT point of (GQP).*

Proof. Step 3 of Algorithm 3 together with Lemma 3.6 ensures that each x^k, y^k, ρ^k, and ω^k, for $k \in \mathcal{K}$, satisfy the conditions of Lemma 3.3. Therefore, (3.6) holds, and by a symmetric argument, $x^k_{\mathcal{A}^*}$ satisfies a similar inequality, and so

$$\|z^k_{\mathcal{I}^*}\| \leq \sqrt{n}\ \omega^k \quad \text{and} \quad \|x^k_{\mathcal{A}^*}\| \leq \sqrt{n}\ \omega^k. \tag{3.11}$$

Note that each x^k, y^k, and z^k satisfies

$$c + Hx^k - A^T y^k = z^k \quad \text{and} \quad y^k = y^{k-1} - \rho^k (Ax^k - b).$$

Rearranging terms, we have

$$\begin{pmatrix} -H_k & A_k^T \\ A_k & \frac{1}{\rho^k} I \end{pmatrix} \begin{pmatrix} x^k_{\mathcal{I}^*} \\ y^k \end{pmatrix} = \begin{pmatrix} [c - z^k]_{\mathcal{I}^*} \\ \frac{1}{\rho^k} y^{k-1} + b \end{pmatrix}. \tag{3.12}$$

Now consider the equality QP (cf. (EQP_k))

$$\underset{x}{\text{minimize}} \quad c^T_{\mathcal{I}^*} x + \tfrac{1}{2} x^T H_k x \quad \text{subject to} \quad A_k x = b,$$

which has optimality conditions

$$\begin{pmatrix} -H_k & A_k^T \\ A_k & \end{pmatrix} \begin{pmatrix} x^*_{\mathcal{I}^*} \\ y^* \end{pmatrix} = \begin{pmatrix} c_{\mathcal{I}^*} \\ b \end{pmatrix}. \tag{3.13}$$

Subtracting (3.12) from (3.13), we get

$$\begin{pmatrix} -H_k & A_k^T \\ A_k & \frac{1}{\rho^k} I \end{pmatrix} \begin{pmatrix} [x^k - x^*]_{\mathcal{I}^*} \\ y^k - y^* \end{pmatrix} = \begin{pmatrix} -z^k_{\mathcal{I}^*} \\ \frac{1}{\rho^k}(y^{k-1} - y^*) \end{pmatrix}. \tag{3.14}$$

Note that this matrix is nonsingular if and only if $\widehat{H}_k := H + \rho^k A_k^T A_k$ is nonsingular [37, Proposition 2]. But Algorithm 2 exits only if $\rho^k > 2\rho_{\min}(\mathcal{A}^k)$, and by Lemma 2.3 and (3.9), $\widehat{H}_k$ is, in fact, positive definite. Therefore, the solutions to (3.13) and (3.14) are unique. Moreover, inverting (3.14), we arrive at

$$\begin{pmatrix} [x^k - x^*]_{\mathcal{I}^*} \\ y^k - y^* \end{pmatrix} = \begin{pmatrix} -\widehat{H}_k^{-1} & \rho^k \widehat{H}_k^{-1} A_k^T \\ \rho^k A_k \widehat{H}_k^{-1} & \rho^k I - (\rho^k)^2 A_k \widehat{H}_k^{-1} A_k^T \end{pmatrix} \begin{pmatrix} -z^k_{\mathcal{I}^*} \\ \frac{1}{\rho^k}(y^{k-1} - y^*) \end{pmatrix}. \tag{3.15}$$

Apply the triangle inequality to the second equation to arrive at

$$\|y^k - y^*\| \le \underbrace{\rho^k \|A_k \widehat{H}_k^{-1}\|}_{(a)} \ \|z^k_{\mathcal{I}^*}\| + \underbrace{\|I - \rho^k A_k \widehat{H}_k^{-1} A_k^T\|}_{(b)} \ \|y^{k-1} - y^*\|. \tag{3.16}$$

Because ρ^k is bounded and $\widehat{H}_k$ is positive definite, there exists a positive constant δ_1 that bounds (a). Next, note that $A_k \widehat{H}_k^{-1} A_k^T = (A_k^{+T} H_k A_k^+ + \rho^k I)^{-1}$. If λ_i are the eigenvalues of $A_k^{+T} H_k A_k^+$, then

$$\rho^k > 2\frac{\max\{0, -\lambda_{\min}(H_k)\}}{\sigma_{\min}(A_k)} > 2 \min_i \lambda_i.$$

Therefore,

$$\|I - \rho^k A_k \widehat{H}_k^{-1} A_k^T\| = \max_i \left(1 - \frac{\rho^k}{\lambda_i + \rho^k}\right) = \max_i \left(\frac{\lambda_i}{\lambda_i + \rho^k}\right) < 1,$$

and so we have a bound on (b). Together with (3.11) and (3.16), this implies that (3.10a) holds.

In order to derive (3.10b), we first observe that (3.11) implies that

$$\|x^k - x^*\| \le \sqrt{n}\ \omega^k + \|[x^k - x^*]_{\mathcal{I}^*}\|. \tag{3.17}$$

Also, from the first set of equations in (3.15),

$$\|[x^k - x^*]_{\mathcal{I}^*}\| \le \|\widehat{H}_k^{-1}\| \ \|z^k_{\mathcal{I}^*}\| + \rho^k \|\widehat{H}_k^{-1} A_k^T\| \ \|y^{k-1} - y^*\|.$$

Substitute (3.10a) into the above, and subsequently substitute the result into (3.17) to obtain (3.10b).

To derive (3.10c), use the definition y^k and the triangle inequality to derive the bound

$$\rho^k \|Ax^k - b\| = \|y^{k-1} - y^k\| \le \|y^k - y^*\| + \|y^{k-1} - y^*\|.$$

Substituting (3.10a) into the above and rearranging terms, we arrive at (3.10c). ∎

It is important to note that the conclusion of Theorem 3.8 does not imply linear convergence of the Lagrange multiplier estimates. However, it still holds that $y^k \to y^*$: repeatedly apply (3.10a) to obtain

$$\|y^{k+\ell} - y^*\| \le \delta_1 \sum_{i=1}^{\ell} \gamma^{\ell - i} \omega^{k+i-1} + \gamma^\ell \|y^k - y^*\|$$

for $\ell \ge 1$. Because $\omega^k \to 0$ and $\gamma < 1$, it follows that $y^{k+\ell} \to y^*$ as $\ell \to \infty$. Thus, $y^k \to y^*$, and with (3.10b) and (3.10c), $x^k \to x^*$ and $\|Ax^k - b\| \to 0$.

4. Finite identification of the active set. An interesting feature of the QPFIL algorithm is its finite identification of the optimal active set: The gradient of the augmented Lagrangian reveals the optimal active set after a finite number of iterations. This key property implies that only finitely many KKT systems need to be solved in order for the algorithm to converge to an exact solution of (GQP). This property is based on the requirement that a gradient-projection step on the augmented Lagrangian subproblem must ensure at least a Cauchy decrease. A strict complementarity assumption is needed.

DEFINITION 4.1 (strict complementarity). *The first-order point* (x^*, y^*) *satisfies strict complementarity if* $[\nabla_x \mathcal{L}_0(x^*, y^*)]_i > 0$ *for all* $j \in \mathcal{A}^*$.

THEOREM 4.2. *Assume that the inner minimization performs a gradient projection that ensures at least a Cauchy decrease on the augmented Lagrangian, that (GQP) satisfies strict complementarity, and that* $(x^k, y^k) \to (x^*, y^*)$*, which is a local minimizer of (GQP). Then Algorithm* 3 *identifies the correct active set in a finite number of iterations.*

Proof. The gradient projection in the inner iteration computes the projected-gradient path and then finds the first minimum of the augmented Lagrangian along this piecewise linear path. The proof largely follows the derivation of the breakpoints and local minima presented in [57, section 16.6].

Because the penalty parameter ρ^k is updated finitely often (see Lemma 3.5), we can assume that it is fixed for all k large enough and that $\rho^k \equiv \rho$. Consider only such k, and let $\nabla\mathcal{L}^k := \nabla_x \mathcal{L}_0(x^k, y^k)$. It follows from the convergence of (x^k, y^k) and the assumption of strict complementarity that, for all k large enough, there exist a neighborhood $\mathcal{N}_\epsilon$ and a positive constant $\tau \gg \epsilon$ such that $(x^k, y^k) \in \mathcal{N}_\epsilon$ and

$$[\nabla\mathcal{L}^k]_i \geq \tau \quad \text{and} \quad x_i^k \leq \epsilon \quad \text{for all } i \in \mathcal{A}^*, \tag{4.1a}$$

$$|[\nabla\mathcal{L}^k]_i| \leq \epsilon \quad \text{and} \quad x_i^k \geq \tau \quad \text{for all } i \in \mathcal{I}^*. \tag{4.1b}$$

We now consider the projected-gradient path for the augmented Lagrangian. The breakpoints along the piecewise linear projected-gradient path from x^k in the direction $-\nabla\,\mathcal{L}^k$ are given by

$$\bar{t}_i = \begin{cases} x_i^k / [\nabla\mathcal{L}^k]_i & \text{if } [\nabla\mathcal{L}^k]_i > 0, \\ \infty & \text{otherwise.} \end{cases}$$

Together with (4.1), this implies that the breakpoints $\bar{t}_i$ satisfy

$$\bar{t}_i \leq \frac{\epsilon}{\tau} \quad \text{for all } i \in \mathcal{A}^* \qquad \text{and} \qquad \bar{t}_i \geq \frac{\tau}{\epsilon} \quad \text{for all } i \in \mathcal{I}^*. \tag{4.2}$$

Because $\tau \gg \epsilon$, it follows that the breakpoints corresponding to active constraints $\mathcal{A}^*$ are much smaller than the breakpoints corresponding to inactive constraints $\mathcal{I}^*$. Note also that if $x_i^k = 0$, then $\bar{t}_i = 0$, and this bound therefore remains active because $[\nabla\mathcal{L}^k]_i \geq \tau > 0$.

The piecewise linear projected-gradient path can now be parameterized in $t \geq 0$:

$$x_i(t) = \begin{cases} x_i^k - t\,[\nabla\mathcal{L}^k]_i & \text{if } t \leq \bar{t}_i, \\ x_i^k - \bar{t}_i [\nabla\mathcal{L}^k]_i & \text{otherwise.} \end{cases}$$

Next, we remove duplicate and zero entries from the breakpoints $\{\bar{t}_1, \ldots, \bar{t}_n\}$ and sort the remaining entries into an ordered sequence

$$0 < t_1 < t_2 < \cdots < t_{a-1} < t_a < t_i < t_{i+1} < \cdots .$$

Observe that (4.2) implies that this ordering separates the active indices $1, \ldots, a$ from the inactive indices $i, i+1, \ldots$, and that

$$t_a \leq \frac{\epsilon}{\tau} \ll \frac{\tau}{\epsilon} \leq t_i.$$

We now show that the first minimizer of the augmented Lagrangian occurs in the interval $[t_a, t_i]$ for ϵ sufficiently small, and therefore the correct active set is identified. We must first demonstrate that the augmented Lagrangian has no minimizer in any of the intervals $[t_{j-1}, t_j]$ for $j \leq a$. Let $j \leq a$, and consider the piecewise search direction on $[t_{j-1}, t_j]$:

$$p_i^{j-1} = \begin{cases} -[\nabla \mathcal{L}^k]_i & \text{if } t_{j-1} \leq \bar{t}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

Next, consider the path segment given by

$$x(t) = x(t_{j-1}) + \Delta t\, p^{j-1} \quad \text{for} \quad \Delta t \in [0, t_j - t_{j-1}],$$

and look for a minimizer of the augmented Lagrangian in this segment. We expand the Lagrangian along this segment and compute the directional gradient on $[t_{j-1}, t_j]$:

$$-f'_{j-1} = -(\nabla \mathcal{L}^k)^T p^{j-1} - x(t_{j-1})^T (H + \rho A^T A) p^{j-1}.$$

From (4.1a) and (4.3), it follows that the first term on the right-hand side above is bounded below by τ^2, whereas the second term is $\mathcal{O}(\epsilon\tau)$. Therefore, the first term dominates, and $-f'_{j-1} \geq \tau^2$ for $j \leq a+1$.

Next, observe that the directional Hessian on $[t_{j-1}, t_j]$,

$$f''_{j-1} = p^{j-1} (H + \rho A^T A) p^{j-1},$$

is bounded because $H + \rho A^T A$ and p^{j-1} are bounded. Using (4.1a) and (4.3), we conclude that there exist positive constants κ_1 and κ_2, independent of ϵ, such that $\kappa_1 \tau^2 \leq f''_{j-1} \leq \kappa_2$. Because $f''_{j-1} > 0$, the minimizer of the augmented Lagrangian on the segment $[t_{j-1}, t_j]$ is given by

$$t_j^* = \min \left\{ \frac{-f'_{j-1}}{f''_{j-1}},\ t_j - t_{j-1} \right\}.$$

The first term in the minimum can be bounded below by $\tau^2/\kappa_2 \gg \epsilon$, and the second term is $\mathcal{O}(\epsilon)$. Thus, for ϵ sufficiently small, the minimum of the quadratic on the segment $[t_{j-1}, t_j]$ occurs at t_j for all $j \leq a$, and the projected-gradient search proceeds to the next segment, $[t_j, t_{j+1}]$. Repeating this argument for all $j \leq a$ shows that there is no minimum of the augmented Lagrangian in any of the segments $[t_{j-1}, t_j]$ for $j \leq a$. Therefore, all active constraints are identified correctly.

Next, we show that the interval $[t_a, t_i]$ contains a minimum of the augmented Lagrangian in its interior. We can use the same estimates as above for the directional gradient and the Hessian because the left-hand boundary corresponds to an active index. Thus, we again consider the case where $f''_a = \mathcal{O}(\kappa_2) > 0$ and $f'_a = \mathcal{O}(\tau^2) > 0$. It follows that the quotient $-f_a/f''_a$ is a constant independent of ϵ. However, the right-hand boundary of the segment

$$\Delta t \in [0,\ t_i - t_a] = [0,\ \tau\mathcal{O}(1/\epsilon)]$$

becomes large as ϵ becomes small. Thus, for ϵ sufficiently small, the minimum of the augmented Lagrangian occurs in the interior of the segment $[t_a, t_i]$, and no additional inactive constraints are identified as active. ∎

Theorem 4.2 implies the following corollary, which establishes global convergence of Algorithm 3.

COROLLARY 4.3. *Let the assumptions of Theorem* 4.2 *hold, and assume in addition that the augmented system* (2.3) *in step* 4 *of Algorithm* 3 *is solved exactly. Then the algorithm terminates finitely at a KKT point of (GQP).*

Proof. The proof follows because Algorithm 3 identifies the correct active set in a finite number of iterations, and an exact solve of (2.3) subsequently gives the solution of (GQP). ∎

5. Numerical results. We implemented the QPFIL algorithm on a subset of medium-scale QPs from the CUTEr [46] test set, with the aim of demonstrating the global convergence and active-set identification properties of QPFIL.

Our test problems are taken from the AMPL versions of the CUTEr test problems [60]; we selected a subset of the test problems that had up to 20,000 variables or constraints. General inequality constraints were converted into equalities by introducing slack variables. The chosen test problems and their sizes are listed in Table 5.1.

The QPFIL algorithm is based on two computational kernels: the gradual minimization of the bound-constrained augmented Lagrangian function (step 3 of Algorithm 2) and the solution of an EQP (step 5 of Algorithm 3). Our implementation uses the bound-constrained solver within TAO [8] (version 1.8.1) which is based on TRON [52] for step 3 of Algorithm 2. TAO's flexible interface allows a user-defined termination criterion; we use this feature to implement the filter-based termination criterion defined in steps 11–12 of Algorithm 2.

The EQP in step 5 of Algorithm 3 is solved by using PETSc [4, 3, 5] (version 2.3.1). We use PETSc's implementation of GMRES with a restart frequency of 300 and an iteration limit of 1000. Other linear solvers can easily be used within the PETSc framework, and specialized linear solvers and preconditioners for KKT systems are currently an active area of research (see, e.g., [58, 44, 31]). We have deliberately chosen the general-purpose solver GMRES because it is readily available within PETSc and because it simplifies our initial implementation.

We compare our implementation with two general-purpose interior-point solvers: KNITRO [18, 62] and LOQO [61]. Although these methods are targeted to general nonlinear optimization problems, both solvers detect whether the problem is a QP and use appropriate algorithmic options. At this stage we are not interested in a direct comparison between QPFIL and these production-quality interior implementations; rather, we are interested in the number of iterations it takes for QPFIL to identify the optimal active set—KNITRO and LOQO serve as relative benchmarks.

All tests were performed on a desktop PC with a 2.5 GHz Intel Pentium 4 processor with 512 KB RAM, running Red Hat Linux version 7.3. Our implementation is compiled using the GNU gcc compiler (version 3.3.5) with the `-O` flag. The source code and makefiles are available from the second author upon request.

We present our numerical results in Table 5.1 comparing major (i.e., outer) iterations and CPU time. The first metric is the most similar across the solvers: at each outer iteration, QPFIL solves a KKT system that is structurally similar to the KKT system that interior-point methods need to solve at each of their own major iterations. The second measure is more dependent on the implementation of the linear algebra. We note that QPFIL easily outperforms the two interior-point methods in terms of

TABLE 5.1
Major iteration counts and CPU times (seconds) for KNITRO, LOQO, and QPFIL.

			KNITRO		LOQO		QPFIL	
Problem	n	m	itns	CPU	itns	CPU	itns	CPU
aug2dc	20200	9996	13	1.9	13	3.5	3	27.5
aug2dcqp	20200	9996	19	4.4	24	3.2	4	85.7
aug2dqp	20192	9996	20	4.6	23	3.1	2	99.2
aug3dcqp	3873	1000	21	0.8	15	0.3	5	0.4
aug3dqp	3873	1000	21	0.8	17	0.3	3	0.5
avgasa	12	6	16	0.0	12	0.0	3	0.0
avgasb	12	6	18	0.0	12	0.0	3	0.0
biggsc4	11	7	17	0.2	21	0.0	5	0.0
blockqp1	2006	1001	21	0.6	15	0.1	2	0.0
blockqp2	2006	1001	21	0.6	10	0.1	1	0.0
blockqp4	2006	1001	21	0.6	16	0.2	8	11.0
cvxqp1	1000	500	16	1.7	25	0.7	18	39.8
dual1	85	1	22	0.2	17	0.1	6	0.0
dual2	96	1	17	0.2	16	0.1	5	0.0
dual3	111	1	16	0.2	16	0.1	6	0.0
dual4	75	1	16	0.1	15	0.1	4	0.0
dualc5	8	1	11	0.0	13	0.0	4	0.0
genhs28	10	8	1	0.0	2	0.0	1	0.0
hatfldh	11	7	13	0.0	16	0.0	2	0.0
hs021	3	1	16	0.0	12	0.0	1	0.0
hs035	4	1	13	0.0	10	0.0	2	0.0
hs044	10	6	6	0.0	15	0.0	6	0.0
hs053	5	3	5	0.0	11	0.0	1	0.0
hs076	7	3	12	0.0	11	0.0	3	0.0
hs118	32	17	22	0.0	17	0.0	10	0.0
hs268	10	5	16	0.0	26	0.0	2	0.0
huestis	10000	2	18	1.2	500	11.0	1	7.9
ksip	1020	1000	17	0.4	42	0.3	17	1.2
liswet11	20002	10000	30	2.3	500	22.9	9	794.0
liswet12	20002	10000	56	4.1	500	22.9	7	680.0
liswet2	20002	10000	31	2.4	10	0.5	12	1200.0
liswet7	20002	10000	27	2.1	11	0.5	8	559.0
lotschd	12	7	12	0.0	19	0.0	1	0.0
mosarqp1	3200	700	18	0.4	15	0.1	7	29.6
mosarqp2	1500	600	15	0.2	19	0.1	6	680.0
ncvxqp5	1000	250	51	2.2	205	11.8	4	4.1
ncvxqp6	1000	250	89	3.6	117	6.9	8	10.7
tame	2	1	2	0.0	9	0.0	1	0.0
zecevic2	4	2	12	0.0	11	0.0	3	0.0

major iterations. This indicates that our strategy for finding the optimal active set is efficient.

In terms of CPU time, the interior-point codes are faster than QPFIL. This disappointing performance of our algorithm can be traced mainly to the poor performance of the GMRES solver on the KKT system (2.3) (the interior-point methods, on the other hand, compute sparse factors of the KKT system). On a typical run, for example, code profiling indicates that the KKT solver consumes almost 90% of CPU time. This situation is discouraging considering the ease with which we are able to solve the bound-constrained subproblem with a conjugate gradient method. We expect

that preconditioners and iterative solvers that are specially designed for KKT systems (see, e.g., [10]) will dramatically improve this performance. We plan to explore this in future implementations.

6. Discussion and conclusions. We have presented a new active-set method for solving QPs that has the potential for solving very large problems and holds the promise of working efficiently on high-performance architectures. We are encouraged by the speed with which the method identifies a correct active set (see Proposition 4.2). One of the remaining challenges for an efficient implementation is finding a computationally effective way to solve the KKT systems arising from the equality-constrained QP subproblems. This is the same problem that must be faced by any interior-point implementation, however; the advantage that we hope to leverage in the QPFIL framework is that the KKT systems are not arbitrarily ill-conditioned.

Two interesting questions remain which we will address in future reports. The first question arises out of Theorem 4.2: Is it possible to simplify the inner iterations further and require gradient-projection steps only until a filter-acceptable point is found? This approach may require a Cauchy-like condition on the inner iteration (which currently is included implicitly by assuming that we perform a few iterations of the minimization of the augmented Lagrangian). Such an approach would have the advantage of removing the need for conjugate gradient iterations involving the Hessian of the augmented Lagrangian—namely, $H + \rho A^T A$—which may be difficult to precondition because of the presence of the term $\rho A^T A$.

The second question concerns the usefulness of the second-order step. If we are far from the minimum, then it may be better to choose the step that adds the largest area to the filter, rather than take a short step in the direction generated by the second-order step. Also, because global convergence relies only on the first-order sequence, it should be possible to save work by solving the KKT systems only approximately, and then tighten the tolerances for the KKT solves when it appears that a correct active set has been identified. We plan to investigate these questions numerically.

Appendix. As discussed in section 3, an alternative to step 6 of Algorithm 2 is to increase the penalty parameter if the current reduced Jacobian is rank deficient. The following theorem, analogous to Theorem 3.4, confirms that the main effect of this increase is that it encourages iterates to move closer to feasibility.

THEOREM A.1 (convergence of inner iterations). *Suppose that step* 6 *of Algorithm* 2 *is replaced by* $\rho^{j+1} \leftarrow 2\rho^j$. *Then the algorithm terminates finitely, or every limit point* x^* *of* $\{x^j\}$ *is a KKT point of (GQP), or it solves* (2.12).

Proof. The case where step 5 tests true finitely many times has already been covered by Theorem 3.4. Therefore, we treat only the cases where step 5 tests true infinitely many times (and thus $\rho^j \to \infty$) and A_* either has full rank or does not.

Case 1 ($\rho^j \to \infty$ and A_* has full rank). Let x^* be any limit point of the subsequence $\{x^j\}_{\mathcal{J}}$. Each x^j, for $j \in \mathcal{J}$, satisfies (2.11), and because $\mathcal{L}_{\rho^j}(x^j, \bar{y}) = \mathcal{L}_0(x^j, y^j)$, Lemma 3.3 holds for the sequences $\{x^j\}$, $\{y^j\}$, $\{\rho^j\}$, and $\{\epsilon^j\}$, where we let $x^k = x^j$, $y^k = y^j$, $\rho^k = \rho^j$, and $\omega^k = \epsilon^j$. Therefore, there exist positive constants α_1 and α_2 such that

$$\|y^j - y^*\| \le \beta^j := \alpha_1 \|x^j - x^*\| + \alpha_2 \epsilon^j \tag{A.1}$$

for all $j \in \mathcal{J}$ large enough, where $y^* := \widehat{y}(x^*)$. Because $\lim_{j \in J} x^j = x^*$ and $\epsilon^j \to 0$, the above implies that $\lim_{j \in J} = y^*$. Hence, (x^*, y^*) satisfies (1.3a).

In order to show that x^* is feasible for (GQP), we use the definition of y^j to derive

$$\rho^j \|Ax^j - b\| = \|y^j - \bar{y}\| \leq \|y^j - y^*\| + \|y^* - \bar{y}\| \leq \beta^j + \|y^* - \bar{y}\|,$$

where we used the triangle inequality and (A.1). Then $\epsilon^j \to 0$, $\lim_{j\in\mathcal{J}} x^j = x^*$, and $\rho^j \to \infty$ imply that $Ax^* = b$. Therefore, x^* satisfies (1.3b), and so (x^*, y^*) is a KKT point of (GQP), as required.

Case 2 ($\rho^j \to \infty$ and A_* does not have full rank). The necessary and sufficient optimality condition for (2.12) is that x^* satisfy

$$\min\{x^*, A^T(Ax^* - b)\} = 0. \tag{A.2}$$

Each x^j and z^j satisfies (2.11). Therefore,

$$\limsup_{j\in J} z^j \equiv c + Hx^* - A^T\bar{y} + \limsup_{j\in J} \rho^j A^T(Ax^j - b) \geq 0,$$

which, because $\rho^j \to \infty$, necessarily implies that

$$z^* := \lim_{j\in J} A^T(Ax^j - b) = A^T(Ax^* - b) \geq 0. \tag{A.3}$$

If we consider only the components $\mathcal{I}^*$ of z^j, then by (2.11) and $\epsilon^j \to 0$, (A.3) holds with equality. But because $\rho^j \to \infty$ and $A^T(Ax^* - b) \geq 0$, we must have that $z^*_{\mathcal{I}} = [A^T(Ax^* - b)]_{\mathcal{I}^*} = 0$. Therefore, x^* and z^* satisfy (A.2). ∎

Acknowledgments. We are grateful to our summer student Anatoly Eydelzon, who developed an initial version of QPFIL using TAO and PETSc, and to Ewout van den Berg for his careful reading of this paper. Sincere thanks to Tamara Kolda, and to two anonymous referees, whose many valuable suggestions helped to clarify the approach and sharpen the theoretical development. In particular, the referees' comments led us to Theorem 3.8 as well as refinements of Algorithm 2.

REFERENCES

[1] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt, *Augmented Lagrangian methods under the constant positive linear dependence constraint qualification*, Math. Program., 111 (2008), pp. 5–32.

[2] M. Anitescu, *Optimization-based simulation of nonsmooth rigid multibody dynamics*, Math. Program., 105 (2006), pp. 113–143.

[3] S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, *PETSc Users Manual (Revision* 2.1.5*)*, Tech. report ANL-95/11, Argonne National Laboratory, Argonne, IL, 2004.

[4] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, *PETSc Web Page*, http://www.mcs.anl.gov/petsc (2001).

[5] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *Efficient management of parallelism in object-oriented numerical software libraries*, in Modern Software Tools for Scientific Computing, E. Arge, A. M. Bruaset, and H. P. Langtangen, eds., Birkhäuser, Boston, 1997, pp. 163–202.

[6] J. L. Barlow and G. Toraldo, *The effect of diagonal scaling on projected gradient methods for bound constrained quadratic programming problems*, Optim. Methods Softw., 5 (1995), pp. 235–245.

[7] G. Bashein and M. Enns, *Computation of optimal controls by a method combining quasilinearization and quadratic programming*, Internat. J. Control, 16 (1972), pp. 177–187.

[8] S. J. Benson, L. Curfman McInnes, J. J. Moré, T. S. Munson, and J. Sarich, *TAO User Manual (Revision* 1.9*)*, Technical manual, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 2007; available online at http://www.mcs.anl.gov/tao.

[9] S. J. Benson, L. Curfman McInnes, J. J. Moré, and J. Sarich, *Scalable algorithms in optimization: Computational experiments*, in Proceedings of the AIAA Multidisciplinary Analysis and Optimization Conference, Albany, NY, 2004.
[10] M. Benzi, G. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.
[11] M. Benzi and G. H. Golub, *A preconditioner for generalized saddle point problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 20–41.
[12] L. Bergamaschi, J. Gondzio, and G. Zilli, *Preconditioning indefinite systems in interior point methods for optimization*, Comput. Optim. Appl., 28 (2004), pp. 149–171.
[13] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.
[14] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.
[15] M. J. Best and J. Kale, *Quadratic programming for large-scale portfolio optimization*, in Financial Services Information Systems, J. Keyes, ed., CRC Press, Boca Raton, FL, 2000, pp. 513–529.
[16] G. Biros and O. Ghattas, *Parallel Lagrange–Newton–Krylov–Schur methods for PDE-constrained optimization. Part* I: *The Krylov–Schur solver*, SIAM J. Sci. Comput., 27 (2005), pp. 687–713.
[17] R. H. Byrd, N. I. M. Gould, J. Nocedal, and R. A. Waltz, *An algorithm for nonlinear optimization using linear programming and equality constrained subproblems*, Math. Program., 100 (2004), pp. 27–48.
[18] R. H. Byrd, J. Nocedal, and R. A. Waltz, *KNITRO: An integrated package for nonlinear optimization*, in Large-Scale Nonlinear Optimization, G. di Pillo and M. Roma, eds., Springer-Verlag, New York, 2006, pp. 35–59.
[19] C. Chin and R. Fletcher, *On the global convergence of an SLP-filter algorithm that takes EQP steps*, Math. Program., 96 (2003), pp. 161–177.
[20] T. F. Coleman and Y. Li, *A reflective Newton method for minimizing a quadratic function subject to bounds on some of the variables*, SIAM J. Optim., 6 (1996), pp. 1040–1058.
[21] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Math. Comp., 50 (1988), pp. 399–430.
[22] A. R. Conn, N. I. M. Gould, and P. L. Toint, *LANCELOT: A Fortran Package for Large-scale Nonlinear Optimization (Release A)*, Springer-Verlag, Heidelberg, Germany, 1992.
[23] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM J. Numer. Anal., 28 (1991), pp. 545–572.
[24] F. Delbos and J. Gilbert, *Global linear convergence of an augmented Lagrangian algorithm for solving convex quadratic optimization problems*, J. Convex Anal., 12 (2005), pp. 45–69.
[25] M. A. Diniz-Ehrhardt, M. A. Gomes-Ruggiero, and S. A. Santos, *Numerical analysis of leaving-face parameters in bound-constrained quadratic minimization*, Optim. Methods Softw., 15 (2001), pp. 45–66.
[26] Z. Dostál, *Box constrained quadratic programming with controlled precision of auxiliary problems and applications*, Z. Angew. Math. Mech., 76 (1996), pp. 413–414.
[27] Z. Dostál, *Box constrained quadratic programming with proportioning and projections*, SIAM J. Optim., 7 (1997), pp. 871–887.
[28] Z. Dostál, A. Friedlander, and S. A. Santos, *Adaptive precision control in quadratic programming with simple bounds and/or equality constraints*, in High Performance Algorithms and Software in Nonlinear Optimization, R. D. Leone, A. Murli, P. M. Pardalos, and G. Toraldo, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998, pp. 161–173.
[29] Z. Dostál, A. Friedlander, and S. A. Santos, *Augmented Lagrangians with adaptive precision control for quadratic programming with equality constraints*, Comput. Optim. Appl., 14 (1999), pp. 37–53.
[30] Z. Dostál, A. Friedlander, and S. A. Santos, *Augmented Lagrangians with adaptive precision control for quadratic programming with simple bounds and equality constraints*, SIAM J. Optim., 13 (2003), pp. 1120–1140.
[31] I. S. Duff, *MA57—A code for the solution of sparse symmetric definite and indefinite systems*, ACM Trans. Math. Software, 30 (2004), pp. 118–144.
[32] K. A. Fegley, S. Blum, J. O. Bergholm, A. J. Calise, J. E. Marowitz, G. Porcelli, and L. P. Sinha, *Stochastic and deterministic design and control via linear and quadratic programming*, IEEE Trans. Automat. Control, 16 (1971), pp. 759–766.
[33] R. Fletcher, *A general quadratic programming algorithm*, J. Inst. Math. Appl., 7 (1971), pp. 76–91.

[34] R. Fletcher, *Resolving degeneracy in quadratic programming*, Ann. Oper. Res., 47 (1993), pp. 307–334.
[35] R. Fletcher and E. Sainz de la Maza, *Nonlinear programming and nonsmooth optimization by successive linear programming*, Math. Programming, 43 (1989), pp. 235–256.
[36] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Math. Program., 91 (2002), pp. 239–269.
[37] A. Forsgren, *Inertia-controlling factorizations for optimization algorithms*, Appl. Numer. Math., 43 (2002), pp. 91–107.
[38] A. Forsgren, P. E. Gill, and M. H. Wright, *Interior methods for nonlinear optimization*, SIAM Rev., 44 (2002), pp. 525–597.
[39] A. Friedlander and J. M. Martínez, *On the maximization of a concave quadratic function with box constraints*, SIAM J. Optim., 4 (1994), pp. 177–192.
[40] M. P. Friedlander and M. A. Saunders, *A globally convergent linearly constrained Lagrangian method for nonlinear optimization*, SIAM J. Optim., 15 (2005), pp. 863–897.
[41] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders, *Preconditioners for indefinite systems arising in optimization*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 292–311.
[42] P. E. Gill, W. Murray, and M. A. Saunders, *User's Guide for SQOPT* 5.3: *A Fortran Package for Large-scale Linear and Quadratic Programming*, Tech. report NA 97-4, Department of Mathematics, University of California, San Diego, 1997.
[43] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *Inertia-controlling methods for general quadratic programming*, SIAM Rev., 33 (1991), pp. 1–36.
[44] G. H. Golub, C. Greif, and J. M. Varah, *An algebraic analysis of a block diagonal preconditioner for saddle point systems*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 779–792.
[45] N. I. M. Gould, S. Leyffer, and Ph. L. Toint, *A multidimensional filter algorithm for nonlinear equations and nonlinear least-squares*, SIAM J. Optim., 15 (2004), pp. 17–38.
[46] N. I. M. Gould, D. Orban, and Ph. L. Toint, *CUTEr and SifDec: A constrained and unconstrained testing environment, revisited*, ACM Trans. Math. Software, 29 (2003), pp. 373–394.
[47] N. I. M. Gould and P. L. Toint, *SQP methods for large-scale nonlinear programming*, in System Modelling and Optimization. Methods, Theory and Applications, M. J. D. Powell and S. Scholtes, eds., Kluwer Academic Publishers, Boston, 2000, pp. 149–178.
[48] N. I. M. Gould and Ph. L. Toint, *An iterative working-set method for large-scale nonconvex quadratic programming*, Appl. Numer. Math., 43 (2002), pp. 109–128.
[49] N. I. M. Gould and P. L. Toint, *Numerical methods for large-scale non-convex quadratic programming*, in Trends in Industrial and Applied Mathematics, A. H. Siddiqi and M. Kocvara, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002, pp. 149–179.
[50] G. D. Hart and M. Anitescu, *A hard-constraint time-stepping approach for rigid multibody dynamics with joints, contact, and friction*, in Proceedings of the 2003 ACM Tapia Conference for Diversity in Computing, Atlanta, GA, 2003, pp. 34–40.
[51] R. M. Larsen, *Combining Implicit Restart and Partial Reorthogonalization in Lanczos Bidiagnalization*, http://sun.stanford.edu/~rmunk/PROPACK/ (2001).
[52] C.-J. Lin and J. J. Moré, *Newton's method for large bound-constrained optimization problems*, SIAM J. Optim., 9 (1999), pp. 1100–1127.
[53] K. Malanowski, *On application of a quadratic programming procedure to optimal control problems in systems described by parabolic equations*, Control Cybernet., 1 (1972), pp. 43–56.
[54] H. M. Markowitz, *The optimization of a quadratic function subject to constraints*, Naval Res. Logist. Quart., 3 (1956), pp. 111–133.
[55] J. J. Moré and G. Toraldo, *On the solution of quadratic programming problems with bound constraints*, SIAM J. Optim., 1 (1991), pp. 93–113.
[56] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, Stud. Appl. Math. 13, SIAM, Philadelphia, 1994.
[57] J. Nocedal and S. Wright, *Numerical Optimization*, Springer-Verlag, Berlin, 1999.
[58] O. Schenk and K. Gärtner, *Solving unsymmetric sparse systems of linear equations with PARDISO*, Future Gen. Comput. Syst., 20 (2004), pp. 475–487.
[59] M. Ulbrich, S. Ulbrich, and L. Vicente, *A globally convergent primal-dual interior-point filter method for nonlinear programming*, Math. Program., 100 (2004), pp. 379–410.
[60] R. Vanderbei, *Benchmarks for Nonlinear Optimization*, http://www.princeton.edu/~rvdb/bench.html (December 2002).
[61] R. Vanderbei and D. Shanno, *An interior point algorithm for nonconvex nonlinear programming*, Comput. Optim. Appl., 13 (1999), pp. 231–252.
[62] R. A. Waltz and T. D. Plantenga, *KNITRO* 5.0 *User's Manual*, Ziena Optimization, Evanston, IL, 2006.