Math. Prog. Comp. (2012) 4:71–107
DOI 10.1007/s12532-012-0035-2

FULL LENGTH PAPER

A primal–dual regularized interior-point method for convex quadratic programs

M. P. Friedlander · D. Orban

Received: 3 September 2010 / Accepted: 1 February 2012 / Published online: 17 February 2012

Abstract Interior-point methods in augmented form for linear and convex quadratic programming require the solution of a sequence of symmetric indefinite linear systems which are used to derive search directions. Safeguards are typically required in order to handle free variables or rank-deficient Jacobians. We propose a consistent framework and accompanying theoretical justification for regularizing these linear systems. Our approach can be interpreted as a simultaneous proximal-point regularization of the primal and dual problems. The regularization is termed *exact* to emphasize that, although the problems are regularized, the algorithm recovers a solution of the original problem, for appropriate values of the regularization parameters.

Mathematics Subject Classification (2000) 90C05 · 90C06 · 90C20 · 90C25 · 90C51 · 65F22 · 65F50

1 Introduction

We consider the primal–dual pair of convex quadratic programs (QPs)

$$\underset{x}{\text{minimize}} \quad c^T x + \tfrac{1}{2} x^T Q x \quad \text{subject to} \quad Ax = b, \quad x \geq 0, \tag{P}$$

M. P. Friedlander (✉)
Department of Computer Science, University of British Columbia, Vancouver, BC, Canada
e-mail: mpf@cs.ubc.ca

D. Orban
GERAD, Department of Mathematics and Industrial Engineering,
École Polytechnique, Montréal, QC, Canada
e-mail: dominique.orban@gerad.ca

$$\underset{x,y,z}{\text{maximize}} \quad b^T y - \tfrac{1}{2} x^T Q x \quad \text{subject to} \quad -Qx + A^T y + z = c, \quad z \geq 0, \qquad \text{(D)}$$

where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $y \in \mathbb{R}^m$, $z \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, and $Q \in \mathbb{R}^{n \times n}$ is symmetric and positive semidefinite; typically $m < n$ although we do not require this assumption. When $Q = 0$, the problems reduce to standard-form linear programs (LPs). These problems are a primal–dual pair in the sense that their optimal objective values are equal and a corresponding optimal primal–dual triple (x, y, z) simultaneously satisfies the Karush–Kuhn–Tucker (KKT) optimality conditions of each problem.

In primal–dual interior-point methods for linear and convex quadratic programming, the computational kernel lies in the solution of a symmetric indefinite system of linear equations that determine each search direction. At each iteration, a saddle-point system—also known as a KKT system—of the form

$$\begin{bmatrix} -(Q+D) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix} \tag{1.1}$$

must be solved; the diagonal matrix D and right-hand side (f_x, f_y) change at each iteration. This system may be solved "as is" using direct or iterative methods suitable for symmetric indefinite systems, or Δx and Δy may be solved for separately via the condensed system

$$A(Q+D)^{-1} A^T \Delta y = A(Q+D)^{-1} f_x + f_y \tag{1.2a}$$

$$(Q+D)\Delta x = A^T \Delta y - f_x. \tag{1.2b}$$

In either case, near rank deficiency of A, or near singularity of $Q + D$, can give rise to inefficient or unstable solutions of these linear systems.

We present an interior-point method based on systematically modifying the linear system (1.1)—or equivalently, (1.2)—in order to alleviate some of these numerical difficulties. We term this technique *exact primal–dual regularization* because the resulting linear systems can be interpreted as coming from an exact regularization of the primal–dual pair (P)–(D). That is, for appropriate parameter values, the regularized problem continues to yield a solution of the unregularized problem. Our approach is closely connected to augmented Lagrangian methods for convex programming, except that it is based on applying a single Newton iteration on each subproblem, rather than on solving it to a specified accuracy.

Our approach is inspired by and shares many characteristics with algorithms that have already been proposed, such as an adaptive diagonal regularization implemented by Altman and Gondzio [1] in the HOPDM [2] software package, and the problem regularization advocated by Saunders and Tomlin [36] and implemented in the PDCO [35] software package. Recently, Bellavia et al. [5] applied similar regularizations to nonnegative linear least-squares problems, and Castro and Cuesta [7] to problems with block-angular constraints. We defer until Sect. 3 a comparative discussion of our approach to previous work.

As a remedy for an ill-conditioned matrix in the (1, 1) block in (1.1) the matrix $Q + D$ can be modified into $Q + D + \rho I$, where $\rho > 0$ is an appropriate regularization

parameter. If the right-hand side remains unaltered, then the new system can be interpreted as the corresponding direction-finding system for the *primal-regularized* QP

$$\begin{array}{ll} \underset{x}{\text{minimize}} & c^T x + \frac{1}{2} x^T Q x + \frac{1}{2}\rho \|x - x_k\|^2 \\ \text{subject to} & Ax = b, \quad x \geq 0, \end{array} \tag{1.3}$$

where x_k is the current iterate; here and throughout, $\|\cdot\|$ is the Euclidian norm.

Similar difficulties arise when A is (nearly) rank deficient, causing factorizations to be numerically unstable and the dual solution to (1.1) to be non-uniquely defined. In that case, a well-defined system can be obtained by replacing the (2, 2) block of (1.1) by δI for an appropriate value of $\delta > 0$. Analogous to (1.3), this can be interpreted as corresponding to the direction finding-system for the *dual-regularized* QP

$$\begin{array}{ll} \underset{x,y,z}{\text{maximize}} & b^T y - \frac{1}{2} x^T Q x - \frac{1}{2}\delta \|y - y_k\|^2 \\ \text{subject to} & -Qx + A^T y + z = c, \quad z \geq 0, \end{array} \tag{1.4}$$

where y_k is the current estimate of the Lagrange multipliers.

A primal–dual regularization amounts to a simultaneous modification of the two diagonal blocks of (1.1). In that case, we arrive at a system in which both the (1, 1) and (2, 2) blocks are modified:

$$K \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix} \quad \text{where} \quad K := \begin{bmatrix} -(Q + D + \rho I) & A^T \\ A & \delta I \end{bmatrix}. \tag{1.5}$$

As we describe in Sect. 2, this amounts to simultaneous regularization of (P) and (D), although the form of the simultaneous primal–dual regularization is different than that given by (1.3)–(1.4). Our approach is built around a perturbation of (P) that amounts to a symmetric regularization of both the primal and dual problems.

These regularization approaches equally benefit the linear algebra of interior-point methods based on solving the condensed system (1.2). In this paper, however, we restrict our attention entirely to the analysis and implementation of an interior method based on modifying the augmented systems. In most cases, similar claims can be made regarding the corresponding condensed systems. However, we prefer the augmented form because it leads to systems that are often sparser; e.g., dense columns in A result in dense matrices in the condensed system. Also, the augmented approach allows for flexibility in how the systems are solved; e.g., Oliveira and Sorensen [28] prove that any preconditioner used for the condensed system has an equivalent for the augmented system; the converse is not true.

The matrix K of (1.5) belongs to the class of symmetric quasi-definite (SQD) matrices and is therefore strongly factorizable [40,41]. In other words, for any permutation matrix P, the indefinite matrix PKP^T possesses a Cholesky-type factorization $L\Delta L^T$ where L is unit lower triangular and Δ is diagonal—without recourse to 2×2 pivoting. This is usually more efficient than Bunch–Parlett-type factorizations [6] because it can be based on a static ordering. The stability of linear systems with SQD matrices

is analyzed by Gill et al. [14]; Saunders [33,34] explores the use of SQD matrices within interior-point methods for linear programming.

In this paper, we present a globally-convergent scheme that identifies a solution to (P) and (D) by means of a primal–dual regularization. Our implementation performs a one-time symbolic factorization of the augmented matrix and, in the linear programming case, a suitable scaling yields an effective condition number that is known in advance and allows for a stable factorization. The main advantages and salient features of our approach are that

(1) sparsity is fully exploited by working directly with the augmented system formulation, and there is no need to make special allowance for dense columns in A, which lead to dense systems (1.2);
(2) dual regularization stabilizes the step computation whenever the constraint matrix A is rank deficient;
(3) primal regularization alleviates the need to transform the problem in the presence of free variables;
(4) the Newton systems are symmetric quasi-definite, leading to an efficient factorization.

Notation

For any given vector $x \in \mathbb{R}^n$, the corresponding capital letter X denotes the $n \times n$ diagonal matrix $\operatorname{diag}(x)$. Let e denote the vector of ones of appropriate length. For any symmetric matrix G, we use $\lambda_{\min}(G)$ and $\lambda_{\max}(G)$ to denote the smallest and largest eigenvalues of G, respectively. Similarly for any matrix H, we use $\sigma_{\min}(H)$ and $\sigma_{\max}(H)$ to denote the smallest and largest singular values of H, respectively.

2 Primal–dual exact regularization

To simplify our exposition, we assume that all variables in (P) have nonnegative constraints. The simple changes required to accomodate other situations are outlined in Sect. 6.

Consider the regularization

$$\begin{array}{ll} \underset{x,r}{\text{minimize}} & c^T x + \frac{1}{2} x^T Q x + \frac{1}{2}\rho\|x - x_k\|^2 + \frac{1}{2}\delta\|r + y_k\|^2 \\ \text{subject to} & Ax + \delta r = b, \quad x \geq 0, \end{array} \tag{2.1}$$

of (P), where $\rho > 0$ and $\delta > 0$ are regularization parameters, and x_k and y_k are current estimates of primal and dual solutions x^* and y^*. Note that $\rho = 0$ and $\delta = 0$ recovers the original problem statement (P). This is a convex quadratic program in the variables (x, r). Its dual is given by

$$\begin{array}{ll} \underset{x,y,s,z}{\text{maximize}} & b^T y - \frac{1}{2} x^T Q x - \frac{1}{2}\delta\|y - y_k\|^2 - \frac{1}{2}\rho\|s + x_k\|^2 \\ \text{subject to} & -Qx + A^T y + z - \rho s = c, \quad z \geq 0, \end{array} \tag{2.2}$$

where y and z are Lagrange multipliers corresponding to the equality and bound constraints of (2.1), respectively. The duality correspondence follows after making the identifications

$$r = y - y_k \quad \text{and} \quad s = x - x_k. \tag{2.3}$$

We note the symmetry of this primal–dual pair: (2.1) contains primal and dual regularization terms, and its dual (2.2) has precisely the same form and also contains corresponding primal and dual regularization terms. The strong symmetry between (2.1) and (2.2) is immediately obvious if we consider the LP case with $Q = 0$, and we see that the problems have symmetric proximal-point terms, and that δr and ρs are the primal and dual residuals, respectively.

These regularized problems are exact—i.e., their solutions solve the original problems (P) and (D)—if $x_k = x^*$ and $y_k = y^*$. Our proposed algorithm (Sect. 4) can be interpreted as applying a single interior-point iteration to a sequence of problems (2.1) and (2.2), where the solution estimates x_k and y_k are updated at each iteration.

3 Related approaches

Regularization approaches for linear and quadratic programming have a long history, and can be broadly categorized into approaches that perturb the original problem and corresponding solution, and those with subproblems that can be interpreted as coming from a perturbed problem, but continue to yield a solution of the original problem. Here we discuss some of the main approaches, each of which hints at some of the desirable properties of (2.1) and (2.2). Note that many of these regularization approaches were developed for linear programming, but can often be extended to quadratic programming.

3.1 Exact regularization

One of the earliest approaches to regularization in optimization is given by Mangasarian and Meyer [21], who show that it is possible to introduce a small perturbation to the objective of an LP, i.e.,

$$\underset{x}{\text{minimize}} \quad c^T x + \rho\phi(x) \quad \text{subject to} \quad Ax = b,\ x \geq 0, \tag{3.1}$$

for some convex perturbation function ϕ, and still obtain a solution of the original problem. Friedlander and Tseng [11] term this property *exact regularization*. The following theorem is due to Mangasarian and Meyer, although the statement appears differently in that paper.

Theorem 3.1 (Mangasarian and Meyer [21]) *Suppose $Q = 0$. Assume* (P) *has a nonempty solution set. Let $\phi : \mathbb{R}^n \to \mathbb{R}$ be a coercive convex differentiable function. Then there exists a $\overline{\rho} > 0$ such that for each $\rho \in (0, \overline{\rho})$, every solution of* (3.1) *is a minimizer of* (P) *and minimizes ϕ over the solution set of* (P).

In essence, Theorem 3.1 states that for small enough perturbations, (primal) solutions of the perturbed LP are also (primal) solutions of the unperturbed LP. Note, however, that the dual solution may be changed. The implication is particularly interesting when an LP has infinitely many solutions; in that case, the regularization term serves to choose exactly *one* of those solutions that also minimizes the perturbation. Obviously, the two-norm regularized LPs (1.3) and (1.4)—with $Q = 0$—satisfy the assumptions of Theorem 3.1. However, if $Q \neq 0$, then the exact regularization property may not hold unless a certain selection problem has a Lagrange multiplier; see Mangasarian and Meyer [21, Theorem 2].

3.2 Proximal-point and augmented Lagrangians

An augmented-Lagrangian approach for treating the equality constraints of (P) is based on solving a sequence of problems of the form

$$\begin{array}{ll} \underset{x,\bar{r}}{\text{minimize}} & c^T x + \frac{1}{2} x^T Q x + \frac{1}{2\delta} \|\bar{r}\|^2 + y_k^T \bar{r} \\ \text{subject to} & Ax + \bar{r} = b, \quad x \geq 0, \end{array} \tag{3.2}$$

where $\delta > 0$ is a penalty parameter and y_k is the current estimate of the vector of Lagrange multipliers associated to the equality constraints [9]. If we introduce $r = (1/\delta)\bar{r}$, then the objective function of (3.2) can be rewritten as

$$c^T x + \tfrac{1}{2} x^T Q x + \tfrac{1}{2} \delta r^T r + \delta r^T y_k,$$

and then (3.2) is equivalent to

$$\begin{array}{ll} \underset{x,r}{\text{minimize}} & c^T x + \frac{1}{2} x^T Q x + \frac{1}{2} \delta \|r + y_k\|^2 \\ \text{subject to} & Ax + \delta r = b, \quad x \geq 0. \end{array} \tag{3.3}$$

Therefore, in view of the discussion related to problem (2.2), we see that (3.2) is a dual regularization of (P). This has been a long-known fact. Rockafellar [30] applies an augmented Lagrangian method to convex programming and observes that it amounts to an application of the proximal-point algorithm, even though the general theory of the proximal-point algorithm was not yet available at the time. The two are tied together by Rockafellar [31]. Note that typically (3.2) and (3.3) are formulated as bound-constrained problems by eliminating the linear constraints.

If (x_{k+1}, r_{k+1}) is a solution to (3.3), the usual first-order multiplier update is

$$y_{k+1} = y_k + \delta^{-1}(b - Ax_{k+1}) = y_k + r_{k+1}. \tag{3.4}$$

Note the resemblance between (3.4) and (2.3). The objective of (3.3) thus penalizes the updated Lagrange multipliers y_{k+1} in the least-squares sense. A classical augmented Lagrangian method solves the sequence of problems (3.3) with increasing accuracy. The parameter δ (often interpreted as a penalty term, rather than a regularization term)

can be changed dynamically, though for convex problems with linear constraints, any positive value of δ is sufficient to induce convergence [10, Sect. 2.3].

Rockafellar [31] considers the sequential minimization of (3.3) (in its bound-constrained formulation) for positive decreasing sequences $\rho_k = \delta_k$ (which need not converge to zero), refers to it as the *proximal method of multipliers* and establishes global convergence properties based on the general proximal-point theory. Wright [42] applies the same framework to linear programs and solves the sequence of problems to within a certain accuracy using a projected gradient method.

3.3 Proximal-point and primal barrier

In the context of a purely primal-barrier algorithm for linear programming, Setiono [37] studies the convergence of a sequence of single Newton iterations applied to a barrier subproblem corresponding to

$$\underset{x}{\text{minimize}} \quad c^T x + \tfrac{1}{2} x^T Q x + \tfrac{1}{2}\rho \|x - x_k\|^2 \quad \text{subject to} \quad Ax = b, \quad x \geq 0; \quad (3.5)$$

here, x_k is the solution estimate of the current barrier subproblem. With an appropriate linesearch, Setiono establishes global and locally linear convergence. Steps are computed from a linear system akin to (1.2). The author reports competitive results when compared with the primal simplex code MINOS 5.3, and claims that his algorithm allowed for an improved residual at the solution. The results of Mangasarian and Meyer [21] imply that a small but positive value of ρ causes the solution of (3.5) to coincide with a solution of (1.3) (with $Q = 0$).

Again in a primal barrier setting for linear programming, Setiono [38] studies a similar algorithm applied to (D), which results in a convex quadratic program. The method uses the formulation

$$\underset{y,z}{\text{minimize}} \quad -\epsilon b^T y + \tfrac{1}{2}\|A^T y + z - c + \epsilon x_k\|^2 \quad \text{subject to} \quad z \geq 0,$$

for different values of $\epsilon > 0$, and requires linear solves with the matrix

$$\begin{bmatrix} AA^T & A \\ A^T & I + \mu_k Z^{-2} \end{bmatrix},$$

where μ_k is the current value of the barrier parameter. While the (2, 2) block of the matrix above allows for free variables in (P), the (1, 1) block can be dense if A has dense columns (a sparse reformulation of linear systems that involve this matrix is possible, however). It appears that Setiono was using the interior-point approach in the more traditional, logarithmic barrier, sense of the term, and not an approach in which the barrier parameter is connected to the duality gap between the primal and dual LPs.

3.4 Primal–dual perturbations

A simultaneous primal–dual regularization for an interior-point approach appears to have been first suggested by Gill et al. [13], and further specialized to LPs by Saunders

and Tomlin [36]. In these approaches, the original problem is perturbed by relaxing the linear constraints and adding Tikhonov regularization terms to the objective:

$$\begin{array}{ll} \underset{x,r}{\text{minimize}} & c^T x + \frac{1}{2} x^T Q x + \frac{1}{2}\rho\|x\|^2 + \frac{1}{2}\|r\|^2 \\ \text{subject to} & Ax + \delta r = b, \; x \geq 0. \end{array} \tag{3.6}$$

Upon applying an interior-point method to (3.6), the linear system solved at each iteration takes the form

$$\begin{bmatrix} -(Q + D + \rho I) & A^T \\ A & \delta I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f_x + \rho x \\ f_y + \delta r \end{bmatrix}, \tag{3.7}$$

in place of (1.1). The primary aim of this approach is to obtain a problem that is always well posed and results in symmetric quasi-definite systems. Gill et al. [14] study the stability of a Cholesky-type factorization applied to (3.7). While closely related to the regularization proposed in this paper, it necessarily perturbs the final solution, and the method does not obtain a solution of the original pair (P)–(D).

3.5 Proximal-point and primal–dual methods

The method described by Altman and Gondzio [1] implemented in the HOPDM [2] software package, and the recent matrix-free version of Gondzio [15], most closely resemble the approach that we describe here. In their approach, they base an interior-point method for QPs on solving a sequence of Newton systems similar to (3.7), except that the right-hand side is not modified; they thus arrive at Newton systems of the form given in (1.5). Altman and Gondzio allow for diagonal perturbations that are not multiples of the identity, but here we simplify and parallel previous discussions by considering only constant-diagonal perturbations. Their approach is based on choosing the diagonal perturbations adaptively, depending on dynamic pivot choices demanded by an indefinite Cholesky factorization routine for symmetric quasi-definite systems. Altman and Gondzio [1] interpret this system as coming from proximal-point regularizations of the pair

$$\begin{array}{lll} \underset{x}{\text{minimize}} & c^T x + \frac{1}{2} x^T Q x + \frac{1}{2}\rho\|x - x_k\|^2 & \text{subject to } Ax = b, \quad x \geq 0, \\ \underset{x,y,z}{\text{maximize}} & b^T y - \frac{1}{2} x^T Q x - \frac{1}{2}\delta\|y - y_k\|^2 & \text{subject to } -Qx + A^T y + z = c, \quad z \geq 0. \end{array}$$

This interpretation is only partly valid because the above problems are not a primal–dual pair, and hence it is not possible to derive their primal–dual interior-point algorithm as coming from these problems. Moreover, their description of the algorithm does not offer a convergence guarantee.

4 A primal–dual regularized interior-point method

The interior-point method that we propose for (P)–(D) contains many of the ingredients common to path-following algorithms. The most notable differences, however,

are the regularization terms that appear in the Newton systems, and the definition of a neighborhood of the central path that accommodates the residual terms r and s in (2.1)–(2.2), which can be interpreted as subproblems to which a single Newton iteration is applied at each step.

4.1 Newton system

Let (x_k, y_k) be temporarily fixed. A primal–dual interior-point method applied to the regularized problems (2.1)–(2.2) is based on applying a single Newton iteration to a sequence of nonlinear equations of the form

$$F_k(w) := \begin{bmatrix} c + Qx + \rho s - A^T y - z \\ \delta(r + y_k) - \delta y \\ \rho x - \rho(s + x_k) \\ Ax + \delta r - b \\ Xz \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \sigma \mu_k e \end{bmatrix}, \quad (x, z) > 0, \tag{4.1}$$

where $w = (x, r, s, y, z)$, $\mu_k > 0$ is the current duality measure (i.e., the *barrier parameter*, which we later define as $x_k^T z_k / n$), and $\sigma \in [0, 1]$ is a *centering parameter*. For fixed ρ_k, δ_k, x_k and y_k, exact solutions $w(\mu_k)$ to (4.1) with $\sigma = 1$ form a smooth trajectory called the *central path* [43]. As $\mu_k \to 0$, it can be shown that this central path leads to a primal–dual solution to (2.1) and (2.2), the necessary and sufficient optimality conditions of which can be succinctly expressed as

$$F_k(w) = 0, \quad \text{and} \quad (x, z) \geq 0.$$

A Newton step for (4.1) from the current iterate w is based on solving the system

$$\begin{bmatrix} Q & 0 & \rho I & -A^T & -I \\ 0 & \delta I & 0 & -\delta I & 0 \\ \rho I & 0 & -\rho I & 0 & 0 \\ A & \delta I & 0 & 0 & 0 \\ Z & 0 & 0 & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta r \\ \Delta s \\ \Delta y \\ \Delta z \end{bmatrix} = - \begin{bmatrix} c + Qx + \rho s - A^T y - z \\ \delta(r + y_k) - \delta y \\ \rho x - \rho(s + x_k) \\ Ax + \delta r - b \\ Xz - \sigma \mu_k e \end{bmatrix}. \tag{4.2}$$

The Jacobian in (4.2) is always nonsingular when $(x, z) > 0$. Note also that, once evaluated at $y = y_k$, the second block equation of (4.2) implies that $\Delta y = r + \Delta r$; this equation is analogous with the Lagrange multiplier update (3.4) typical of augmented Lagrangian methods.

Reducing (4.2) by eliminating the variables Δr and Δs, we obtain

$$\begin{bmatrix} -(Q + \rho I) & A^T & I \\ A & \delta I & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} c + Qx + \rho(x - x_k) - A^T y - z \\ b - Ax - \delta(y - y_k) \\ \sigma \mu_k e - Xz \end{bmatrix}. \tag{4.3}$$

In our approach we take $x = x_k$ and $y = y_k$ at each iteration, and so upon further simplification and elimination of Δz we arrive at the system

$$\begin{bmatrix} -(Q + X^{-1}Z + \rho I) & A^T \\ A & \delta I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} c + Qx - A^T y - \sigma \mu_k X^{-1} e \\ b - Ax \end{bmatrix}, \quad (4.4)$$

where the remaining search directions are recovered via

$$\Delta r = -r + \Delta y, \quad (4.5a)$$
$$\Delta s = -s + \Delta x, \quad (4.5b)$$
$$\Delta z = c + Qx - A^T y - z + (Q + \rho I)\Delta x - A^T \Delta y. \quad (4.5c)$$

The system (4.4) is used in our convergence analysis of Sect. 5. Note that the matrix of (4.4) has precisely the form (1.5), and that its sparsity is entirely determined by Q and A because the modifications to the (1, 1) and (2, 2) blocks are diagonal.

4.2 A long-step interior method

Path-following methods are based on generating iterates that remain within a neighborhood of the central path. We define a neighborhood $\mathcal{N}_k$ of the central path as the set of points (x, r, s, y, z) that satisfy a specific subset of the conditions

$$\bar{\gamma}_{\text{C}}\, x^T z/n \geq [x]_i [z]_i \geq \gamma_{\text{C}}\, x^T z/n, \quad (4.6a)$$
$$x^T z \geq \gamma_{\text{P}} \|Ax + \delta_k r - b\|, \quad (4.6b)$$
$$x^T z \geq \gamma_{\text{D}} \|c + Qx + \rho_k s - A^T y - z\|, \quad (4.6c)$$
$$x^T z \geq \gamma_{\text{R}} \|\delta_k (r + y_k) - \delta_k y\|, \quad (4.6d)$$
$$x^T z \geq \gamma_{\text{S}} \|\rho_k x - \rho_k (s + x_k)\|, \quad (4.6e)$$

where $0 < \gamma_{\text{C}} < 1 < \bar{\gamma}_{\text{C}}$ and $(\gamma_{\text{P}}, \gamma_{\text{D}}, \gamma_{\text{R}}, \gamma_{\text{S}}) > 0$ are given constants. Note the dependence of $\mathcal{N}_k$ on ρ_k, δ_k, x_k and y_k. In the following we give two algorithms, each of which uses a fixed subset of the conditions (4.6).

If we assume that (P) has a strictly interior feasible point, it possesses a primal–dual solution and we expect in this case that $\delta_k r_k \to 0$ and $\rho_k s_k \to 0$. The algorithms we have in mind follow traditional interior-point methods for convex programming and drive the duality gap $x^T z$ to zero. However, conditions such as $\|\delta r\| = O(x^T z)$ and $\|\rho s\| = O(x^T z)$, as suggested by (4.6d) and (4.6e), will ultimately be impossible to satisfy on infeasible problems. We therefore use such conditions as a mechanism for detecting infeasibility: if $x^T z$ becomes "small" but δr remains "large", then (P) is declared infeasible. Similarly, if $x^T z$ becomes "small" but ρs remains "large", (D) is declared infeasible. These intentionally vague statements are made more precise in what follows.

Our interior-point scheme generates the next iterate w_{k+1} as follows. We compute the step Δw from (4.4) and (4.5), and a steplength $\alpha_k \in (0, 1]$ such that

$$w_k(\alpha_k) \in \mathcal{N}_{k+1} \quad (4.7)$$

Algorithm 4.1 Primal–dual regularized interior-point algorithm

Step 0. **[Initialize]** Choose minimum and maximum centering parameters $0 < \sigma_{\min} \leq \sigma_{\max} < 1$, a constant $\sigma_{\max} < \beta < 1$, proximity parameters $0 < \gamma_C < 1 < \bar{\gamma}_C$, $(\gamma_P, \gamma_D, \gamma_R, \gamma_S) > 0$, initial regularization parameters $\rho_0 > 0$, $\delta_0 > 0$, and a stopping tolerance $\epsilon > 0$. Choose initial primal $x_0 \in \mathbb{R}^n_{++}$, $r_0 \in \mathbb{R}^m$ and dual guesses $s_0 \in \mathbb{R}^n$, $y_0 \in \mathbb{R}^m$, $z_0 \in \mathbb{R}^n_{++}$ so that $w_0 \in \mathcal{N}_0$. Set $\mu_0 = x_0^T z_0/n$ and $k = 0$.

Step 1. **[Test convergence]** If $x_k^T z_k \leq \epsilon$, declare convergence.

Step 2. **[Step computation]** Choose a centering parameter $\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$. Compute the Newton step $\Delta w_k = (\Delta x_k, \Delta r_k, \Delta s_k, \Delta y_k, \Delta z_k)$ from w_k by solving

$$\begin{bmatrix} -(Q + X_k^{-1} Z_k + \rho_k I) & A^T \\ A & \delta_k I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} c + Qx_k - A^T y_k - \sigma_k \mu_k X_k^{-1} e \\ b - Ax_k \end{bmatrix} \tag{4.8}$$

and recovering Δr_k, Δs_k and Δz_k from (4.5).

Step 3. **[Linesearch]** Select $\delta_{k+1} \in (0, \delta_k]$ and $\rho_{k+1} \in (0, \rho_k]$ and compute α_k as the largest $\alpha \in (0, 1]$ such that

$$w_k(\alpha) \in \mathcal{N}_{k+1} \quad \text{and} \quad \mu_k(\alpha) \leq (1 - \alpha[1 - \beta])\mu_k, \tag{4.9}$$

where $w_k(\alpha) = w_k + \alpha \Delta w_k$ and $\mu_k(\alpha) = x_k(\alpha)^T z_k(\alpha)/n$.

Step 4. **[Update iterates]** Set $w_{k+1} = w_k(\alpha_k)$, $\mu_{k+1} = \mu_k(\alpha_k)$. Increment k by 1 and go to Step 1.

where

$$w_k(\alpha_k) := (x_k + \alpha_k \Delta x, r_k + \alpha_k \Delta r, s_k + \alpha_k \Delta s, y_k + \alpha_k \Delta y, z_k + \alpha_k \Delta z).$$

Note that the definition of $\mathcal{N}_{k+1}$ now involves ρ_{k+1} and δ_{k+1}. The value of those parameters must be selected together with α_k to ensure that (4.7) is satisfied.

A full description of our interior-point method is formally outlined as Algorithm 4.1. Note that (4.8) in Step 2 is simply (4.4) evaluated at w_k. We leave the updates of the regularization parameters ρ_k and δ_k in Step 3 intentionally vague; these updates are made precise in the specialized algorithms 5.1, 5.2 and 5.3.

In their long-step method, Kojima et al. [20] use a test of the form $\|(x_k, z_k)\| > \omega$ to detect infeasible problems. In their analysis, if Algorithm 4.1 stops because $\|(x_k, z_k)\| > \omega$, then (P) and/or (D) is locally infeasible. In the present framework, we expect that local infeasibility will be characterized by either or both of $\|\delta_k r_k\|$ and $\|\rho_k s_k\|$ failing to converge to zero. This intuition will be confirmed by the convergence analysis.

5 Convergence analysis

We begin with an analysis of the components of Algorithm 4.1 that do not depend on how the regularization parameters are updated. In Sects. 5.2 and 5.3 we provide convergence analyses for two particular strategies (Algorithms 5.1 and 5.2) for updating the regularization parameters. Each strategy uses a different subset of conditions (4.6) to define a neighborhood $\mathcal{N}_k$. Our analysis of both cases depends on the assumption that the sequence of steplengths $\{\alpha_k\}$ is bounded away from zero; it may be possible to avoid this assumption—as per the analysis in Wright [43, Ch. 6]—but the additional slack variables r and s in our formulation complicate the subsequent analysis.

5.1 Global convergence: preliminaries

We begin our analysis by providing bounds on the eigenvalues of the matrix

$$K_k := \begin{bmatrix} -H_k & A^T \\ A & \delta_k I \end{bmatrix} \quad \text{where} \quad H_k := Q + X_k^{-1} Z_k + \rho_k I,$$

which is the matrix that appears in Step 2 of Algorithm 4.1. In the remainder of this section, we simplify the notation by dropping the iteration counter k.

The eigenvalue bounds

$$\lambda_{\min}(H) \geq \lambda_{\min}(Q) + \min_{1 \leq i \leq n} \frac{z_i}{x_i} + \rho \quad \text{and} \quad \lambda_{\max}(H) \leq \lambda_{\max}(Q) + \max_{1 \leq i \leq n} \frac{z_i}{x_i} + \rho,$$

and congruence relation

$$\begin{bmatrix} -H & A^T \\ A & \delta I \end{bmatrix} = \begin{bmatrix} I & 0 \\ -AH^{-1} & I \end{bmatrix} \begin{bmatrix} -H & 0 \\ 0 & AH^{-1}A^T + \delta I \end{bmatrix} \begin{bmatrix} I & -H^{-1}A^T \\ 0 & I \end{bmatrix}, \tag{5.1}$$

are useful for the following result, which is inspired by Rusten and Winther [32, Lemma 2.1] and Silvester and Wathen [39, Lemma 2.2]. Gondzio [15] gives a similar result for the case where A has full rank.

Theorem 5.1 *For all $(x, z) > 0$ and all $\rho > 0$ and $\delta > 0$, K possesses precisely n negative eigenvalues and m positive eigenvalues. Let them be denoted and ordered as*

$$\lambda_{-n} \leq \lambda_{-n+1} \leq \cdots \leq \lambda_{-1} < 0 < \lambda_1 \leq \cdots \leq \lambda_{m-1} \leq \lambda_m.$$

The largest positive and smallest negative eigenvalues of K satisfy the bounds

$$\lambda_{-n} \geq \tfrac{1}{2}[\delta - \lambda_{\max}(H)] - \tfrac{1}{2}\left[(\lambda_{\max}(H) - \delta)^2 + 4\left(\sigma_{\max}(A)^2 + \lambda_{\max}(H)\delta\right)\right]^{1/2}, \tag{5.2a}$$

$$\lambda_m \leq \tfrac{1}{2}[\delta - \lambda_{\min}(H)] + \tfrac{1}{2}\left[(\lambda_{\min}(H) - \delta)^2 + 4\left(\sigma_{\max}(A)^2 + \lambda_{\min}(H)\delta\right)\right]^{1/2}. \tag{5.2b}$$

The smallest positive and largest negative eigenvalues of K satisfy the bounds

$$\lambda_{-1} \leq -\lambda_{\min}(H), \tag{5.3a}$$

$$\lambda_1 \geq \tfrac{1}{2}[\delta - \lambda_{\max}(H)] + \tfrac{1}{2}\left[(\lambda_{\max}(H) - \delta)^2 + 4\left(\sigma_{\min}(A)^2 + \lambda_{\max}(H)\delta\right)\right]^{1/2}. \tag{5.3b}$$

Moreover, $\lambda_1 = \delta$ is the smallest positive eigenvalue of K if and only if A does not have full row rank. In this case, its associated eigenspace is $\{0\} \times Null(A^T)$. Its geometric multiplicity is thus $m - rank(A)$.

Proof The first part of the theorem follows from (5.1) and Sylvester's law of inertia. Indeed, note that H and $AH^{-1}A^T + \delta I$ are positive definite because of the assumed positivity of x, z, ρ, and δ.

The rest of the proof parallels Silvester and Wathen [39, Lemma 2.2]. If $(u, v) \neq (0, 0)$ is an eigenvector of K associated to the eigenvalue λ, then

$$-Hu + A^T v = \lambda u, \tag{5.4a}$$

$$Au + \delta v = \lambda v. \tag{5.4b}$$

Note that $\lambda = \delta > 0$ is an eigenvalue of K if and only if A does not have full row rank, and in this case, its associated eigenspace is $\{0\} \times \text{Null}(A^T)$.

Now suppose $\lambda \neq \delta$. From (5.4b), we have $v = (\lambda - \delta)^{-1} Au$. Necessarily, $u \neq 0$. Substituting into (5.4a) and taking the inner product with u yields

$$\lambda \|u\|^2 = -u^T H u + (\lambda - \delta)^{-1} u^T A^T A u. \tag{5.5}$$

If $\lambda < \delta$, then, because the right-hand side of (5.5) is negative, we must have $\lambda < 0$, and so $\lambda_1 \geq \delta$. But by the implication drawn from (5.4), $\lambda_1 = \delta$ if and only if A is rank deficient. Hence, that last statement of the theorem is proved.

If $\lambda > \delta$, we deduce from (5.5) that

$$\lambda \|u\|^2 \leq -\lambda_{\min}(H)\|u\|^2 + (\lambda - \delta)^{-1} \sigma_{\max}(A)^2 \|u\|^2.$$

Upon simplifying and substituting ℓ for λ, we see that the quadratic

$$\ell^2 + (\lambda_{\min}(H) - \delta)\ell - (\sigma_{\max}(A)^2 + \lambda_{\min}(H)\delta)$$

in ℓ takes a nonpositive value when evaluated at an eigenvalue $\lambda > \delta$ of K. In particular this must be true of λ_m, which yields (5.2b).

If $\lambda < 0$, (5.5) yields the bound

$$\lambda \|u\|^2 \geq -\lambda_{\max}(H)\|u\|^2 + (\lambda - \delta)^{-1} \sigma_{\max}(A)^2 \|u\|^2.$$

In turn, this implies that the quadratic in ℓ,

$$\ell^2 + (\lambda_{\max}(H) - \delta)\ell - (\sigma_{\max}(A)^2 + \lambda_{\max}(H)\delta),$$

takes a nonpositive value when evaluated at an eigenvalue $\lambda < 0$ of K. In particular this must be true of λ_{-n}, which yields (5.2a).

We now establish (5.3a). Taking the inner product of (5.4a) with u and (5.4b) with v and subtracting, we obtain

$$(\lambda - \delta)\|v\|^2 = u^T H u + \lambda \|u\|^2 \geq \lambda_{\min}(H)\|u\|^2 + \lambda \|u\|^2.$$

For all $\lambda < 0$, the left-hand side of this last inequality is nonpositive and thus, $\lambda \leq -\lambda_{\min}(H)$. In particular, this proves (5.3a).

Having established earlier that $\delta > 0$ was the smallest positive eigenvalue of K if and only if A does not have full row rank, we may assume without loss of generality for the last part that A has full row rank. Extracting u from (5.4a) gives $u = (H + \lambda I)^{-1} A^T v$. Necessarily, $v \neq 0$. Injecting u into (5.4b) and taking the inner product with v reveals that

$$(\lambda - \delta)\|v\|^2 = v^T A(H + \lambda I)^{-1} A^T v \geq (\lambda_{\max}(H) + \lambda)^{-1} \sigma_{\min}(A)^2 \|v\|^2.$$

Dividing through by $\|v\|^2$ and rearranging terms, we thus require the quadratic

$$\ell^2 + (\lambda_{\max}(H) - \delta)\ell - (\sigma_{\min}(A)^2 + \lambda_{\max}(H)\delta)$$

in ℓ to be nonnegative, which is true when evaluated at an eigenvalue larger than its right-most root. In particular, this is true of λ_1, and proves (5.3b). □

The following corollary to Theorem 5.1 is central to our approach. It implies that the inverse of the matrix defining the Newton system in (4.8) is uniformly bounded in norm for given positive regularization parameters ρ and δ, and that the bound can be directly controlled by these parameters. The result follows from the observation that (5.3a)–(5.3b) imply $\lambda_{-1} \leq -\rho$ and $\lambda_1 \geq \delta$.

Corollary 5.2 *For all $(x, z) > 0$ and all $\rho > 0$ and $\delta > 0$,*

$$\|K^{-1}\| \leq 1/\min(\rho, \delta). \tag{5.6}$$

We will make use of the following inequalities in the sections to come. Let $w(\alpha) = w + \alpha \Delta w$. Turning now to the progress achieved along the Newton direction, we have

$$\begin{aligned} Ax(\alpha) + \delta r(\alpha) - b &= (Ax + \delta r - b) + \alpha(A\Delta x + \delta \Delta r) \\ &= (1 - \alpha)(Ax + \delta r - b), \end{aligned} \tag{5.7}$$

and

$$\begin{aligned} &c + Qx(\alpha) + \rho s(\alpha) - A^T y(\alpha) - z(\alpha) \\ &\quad = (c + Qx + \rho s - A^T y - z) + \alpha(Q\Delta x + \rho \Delta s - A^T \Delta y - \Delta z) \\ &\quad = (1 - \alpha)(c + Qx + \rho s - A^T y - z). \end{aligned}$$

Similarly, because the Newton step is computed from the current iterate w_k,

$$\delta r(\alpha) = (1 - \alpha)\delta r + \alpha \delta \Delta y \tag{5.8}$$

and

$$\rho s(\alpha) = (1 - \alpha)\rho s + \alpha \rho \Delta x.$$

Additionally, (4.2) yields

$$\begin{aligned} z^T\Delta x + x^T\Delta z &= -x^Tz + n\sigma\mu \\ &= -x^Tz + \sigma x^Tz \\ &= -(1-\sigma)x^Tz, \end{aligned} \tag{5.9}$$

and

$$z_i[\Delta x]_i + x_i[\Delta z]_i = -x_iz_i + \sigma x^Tz/n. \tag{5.10}$$

5.2 First variation: fixed regularization

Our first method, described in Algorithm 5.1, holds the regularization parameters ρ and δ fixed at all iterations and enforces conditions (4.6a), (4.6b) and (4.6c) at each iteration.

Algorithm 5.1 Variation of the primal–dual method with constant regularization

Apply Algorithm 4.1 with $\rho_k := \rho_0 > 0$ and $\delta_k := \delta_0 > 0$ for all k. In Step 3, only conditions (4.6a), (4.6b), and (4.6c) are enforced.

Because the regularization parameters are constant in this section, we simply denote them ρ and δ for readability. Convergence properties rely on the following technical lemma.

Lemma 5.3 *Suppose that* $(\Delta x, \Delta y, \Delta z)$ *is given by Step 2 of Algorithm 5.1, and the sequence* $\{(r_k, s_k, z_k)\}$ *is bounded. Then there exists a constant* π*, dependent only on* n*, such that*

$$|\Delta x_i \Delta z_i| \le \pi, \quad |\Delta x_i \Delta z_i - \gamma_c \Delta x^T \Delta z/n| \le \pi, \quad \textit{and} \quad |\Delta x_i \Delta z_i - \bar{\gamma}_c \Delta x^T \Delta z/n| \le \pi.$$

Proof In order to prove the required result, it is sufficient to demonstrate that $(\Delta x, \Delta y, \Delta z)$ is bounded. To that end, we first show that $(\Delta x, \Delta y)$ is bounded, and second show that Δz is bounded.

We have from (4.6b) and (4.9) that

$$\begin{aligned} \|Ax_k - b\| &\le \|Ax_k + \delta r_k - b\| + \delta\|r_k\| \le x_k^T z_k / \gamma_{\text{P}} + \delta \sup_k \|r_k\| \\ &\le x_0^T z_0 / \gamma_{\text{P}} + \delta \sup_k \|r_k\|, \end{aligned}$$

which shows that the second block of the right-hand side in (4.8) is bounded.

We now show that the first block in (4.8) is bounded. It follows from (4.6a) that

$$\frac{\sigma}{\bar{\gamma}_{\text{C}}} z \le \sigma\mu X^{-1}e \le \frac{\sigma}{\gamma_{\text{C}}} z$$

componentwise. As a consequence,

$$\|\sigma_k \mu_k X_k^{-1} e - z_k\| \leq M \sup_k \|z_k\|, \qquad \text{where} \quad M := \max\left(\left|\frac{\sigma}{\gamma_C} - 1\right|, \left|\frac{\sigma}{\bar{\gamma}_C} - 1\right|\right).$$

Combining this last inequality with (4.6c) and (4.9), we obtain

$$\begin{aligned}\|c + Qx_k - A^T y_k - \sigma_k \mu_k X_k^{-1} e\| &= \|(c + Qx_k + \rho s_k - A^T y_k - z) - \rho s_k \\ &\quad - (\sigma_k \mu_k X_k^{-1} e - z_k)\| \\ &\leq x_0^T z_0 / \gamma_D + \rho \sup_k \|s_k\| + M \sup_k \|z_k\|.\end{aligned}$$

By Corollary 5.2, the inverse of the matrix in (4.8) is bounded, and so $(\Delta x, \Delta y)$ is bounded because the right-hand side of that equation is bounded. To show that Δz is bounded, we note that Δz satisfies Eq. (4.5c), which has a bounded right-hand side. □

We are now in position to establish the convergence properties of Algorithm 5.1. The next result implies that the duality measure μ_k converges to zero under a boundedness assumption.

Theorem 5.4 *Suppose that Algorithm 5.1 with $\epsilon = 0$ generates the sequence $\{w_k\}$, and that the sequence $\{(r_k, s_k, z_k)\}$ is bounded. Then $\{\mu_k\} \to 0$.*

Proof We follow Kojima et al. [20], and express (4.9) in Step 3 of the algorithm as

$$(f_i(\alpha), \bar{f}_i(\alpha)) \geq 0, \quad h(\alpha) \geq 0, \quad g_P(\alpha) \geq 0, \quad \text{and} \quad g_D(\alpha) \geq 0, \tag{5.11}$$

for $i = 1, \ldots, n$, where, dropping for the moment the subscript k counter,

$$f_i(\alpha) := (x_i + \alpha \Delta x_i)(z_i + \alpha \Delta z_i) - \gamma_C (x + \alpha \Delta x)^T (z + \alpha \Delta z)/n, \tag{5.12a}$$

$$\bar{f}_i(\alpha) := \bar{\gamma}_C (x + \alpha \Delta x)^T (z + \alpha \Delta z)/n - (x_i + \alpha \Delta x_i)(z_i + \alpha \Delta z_i), \tag{5.12b}$$

$$h(\alpha) := (1 - \alpha(1 - \beta)) x^T z - (x + \alpha \Delta x)^T (z + \alpha \Delta z), \tag{5.12c}$$

$$g_P(\alpha) := (x + \alpha \Delta x)^T (z + \alpha \Delta z) - \gamma_P \|A(x + \alpha \Delta x) + \delta(r + \alpha \Delta r) - b\|, \tag{5.12d}$$

$$\begin{aligned} g_D(\alpha) := &(x + \alpha \Delta x)^T (z + \alpha \Delta z) \\ &- \gamma_D \|c + Q(x + \alpha \Delta x) + \rho(s + \alpha \Delta s) - A^T (y + \alpha \Delta y) - (z + \alpha \Delta z)\|, \end{aligned} \tag{5.12e}$$

and g_P and g_D have been suitably modified to account for the regularization terms and Qx, the gradient of the quadratic term. By way of contradiction, we assume that $x^T z \geq \bar{\epsilon}$ for some $\bar{\epsilon} > 0$. Exactly as in Kojima et al. [20, Sect. 3], we use (5.9)–(5.10)

and Lemma 5.3, and the inequality $x^T z \geq \bar{\epsilon}$ to show that $f_i(\alpha) \geq 0$, $\bar{f}_i(\alpha) \geq 0$ and $h(\alpha) \geq 0$ for all

$$0 \leq \alpha \leq \min\left\{1, \frac{(1-\gamma_{\mathrm{C}})\sigma_{\min}\bar{\epsilon}}{n\pi}, \frac{(\bar{\gamma}_{\mathrm{C}}-1)\sigma_{\min}\bar{\epsilon}}{n\pi}, \frac{(\beta-\sigma_{\max})\bar{\epsilon}}{\pi}\right\}.$$

We now establish a similar property for $g_{\mathrm{P}}(\alpha)$ and $g_{\mathrm{D}}(\alpha)$. Using (5.7), (5.9) and (5.12),

$$\begin{aligned} g_{\mathrm{P}}(\alpha) &= (1-\alpha)x^T z + \alpha\sigma x^T z + \alpha^2 \Delta x^T \Delta z - (1-\alpha)\gamma_{\mathrm{P}}\|Ax + \delta r - b\| \\ &= (1-\alpha)(x^T z - \gamma_{\mathrm{P}}\|Ax + \delta r - b\|) + \alpha\sigma x^T z + \alpha^2 \Delta x^T \Delta z \\ &\geq \alpha\sigma\bar{\epsilon} - \alpha^2\pi, \end{aligned}$$

where the last inequality uses (4.6b), Lemma 5.3, and the assumption that $x^T z \geq \bar{\epsilon}$. This last expression is nonnegative for all $\alpha \in [0, \sigma\bar{\epsilon}/\pi]$. Similarly, $g_{\mathrm{D}}(\alpha) \geq 0$ for all $\alpha \in [0, \sigma\bar{\epsilon}/\pi]$.

Upon setting

$$\alpha^* := \min\left\{1, \frac{(1-\gamma_{\mathrm{C}})\sigma_{\min}\bar{\epsilon}}{\pi}, \frac{(\bar{\gamma}_{\mathrm{C}}-1)\sigma_{\min}\bar{\epsilon}}{\pi}, \frac{(\beta-\sigma_{\max})\bar{\epsilon}}{\pi}, \frac{\sigma_{\min}\bar{\epsilon}}{\pi}\right\},$$

and noting that Step 3 of Algorithm 4.1 selects the largest $\alpha \in (0, 1]$ satisfying (5.11), we deduce that $\alpha_k \geq \alpha^*$ for all k. But then the condition $\mu_{k+1} \leq (1 - \alpha_k(1-\beta))\mu_k$ implies that

$$0 < \bar{\epsilon} \leq x_{k+1}^T z_{k+1} \leq (1 - \alpha^*[1-\beta])x_k^T z_k \leq \cdots \leq (1 - \alpha^*[1-\beta])^{k+1} x_0^T z_0.$$

This is a contradiction because the rightmost term above converges to zero. □

Establishing global convergence of the algorithm is now a matter of examining the nature of the limit points of $\{r_k\}$ and $\{s_k\}$.

Theorem 5.5 *Suppose Algorithm 5.1 with $\epsilon = 0$ generates the sequence $\{w_k\}$, and that $\{(r_k, s_k, z_k)\}$ remains bounded. Then if r_* and s_* denote particular limit points of $\{r_k\}$ and $\{s_k\}$ defined by subsequences indexed by the index set $\mathcal{K} \subseteq \mathbb{N}$, every limit point of $\{(x_k, z_k)\}_{\mathcal{K}}$ determines a primal–dual solution of the primal–dual pair*

$$\underset{x}{\text{minimize}}\ (c + \rho s_*)^T x + \tfrac{1}{2}x^T Q x \quad \text{subject to } Ax = b - \delta r_*,\ x \geq 0, \tag{5.13}$$

$$\underset{x,y,z}{\text{maximize}}\ (b - \delta r_*)^T y - \tfrac{1}{2}x^T Q x \quad \text{subject to } -Qx + A^T y + z = c + \rho s_*,\ z \geq 0. \tag{5.14}$$

Proof We may assume, by reducing to a further subsequence if necessary, that $\{(x_k, z_k)\}_{\mathcal{K}} \to (\bar{x}, \bar{z})$. As in the proof of Lemma 5.3, the dual residual $(c + Qx_k - A^T y_k - z_k)$

remains bounded. Therefore, reducing to another subsequence if necessary, we may assume that $\{A^T y_k\}_{\mathcal{K}} \to A^T \bar{y}$ for some $\bar{y}$. In the limit along $\mathcal{K}$, we now have

$$c + Q\bar{x} + \rho s_* - A^T \bar{y} - \bar{z} = 0, \quad A\bar{x} + \delta r_* = b, \quad \bar{X}\bar{z} = 0, \quad (\bar{x}, \bar{z}) \geq 0,$$

which are the necessary and sufficient optimality conditions of (5.13)–(5.14). This follows from the neighborhood conditions (4.6a)–(4.6c), together with $\{\mu_k\} \to 0$, which follows from Theorem 5.4. □

The next result relates some limit points of $\{r_k\}$ and $\{s_k\}$ with feasibility of (P)–(D).

Theorem 5.6 *Suppose Algorithm 5.1 with $\epsilon = 0$ generates the sequence $\{w_k\}$, and that $\{(r_k, s_k, z_k)\}$ remains bounded. Then*

(1) *If $\liminf_{k\in\mathbb{N}} \|r_k\| = 0$, every limit point of $\{x_k\}_{\mathcal{K}}$ is feasible for (P), where $\mathcal{K} \subseteq \mathbb{N}$ is an index set such that $\{r_k\}_{\mathcal{K}} \to 0$.*
(2) *If $\liminf_{k\in\mathbb{N}} \|s_k\| = 0$, every limit point of $\{(x_k, z_k)\}_{\mathcal{K}'}$ determines a feasible point for (D), where $\mathcal{K}' \subseteq \mathbb{N}$ is an index set such that $\{s_k\}_{\mathcal{K}'} \to 0$.*
(3) *If there exists an index set $\mathcal{K}'' \subseteq \mathbb{N}$ such that $\{r_k\}_{\mathcal{K}''} \to 0$ and $\{s_k\}_{\mathcal{K}''} \to 0$, every limit point $(\bar{x}, \bar{z})$ of $\{(x_k, z_k)\}_{\mathcal{K}''}$ determines a primal–dual solution of (P)–(D).*

Proof We first establish part 1. By assumption there is an index set $\mathcal{K} \subseteq \mathbb{N}$ such that $\{r_k\}_{\mathcal{K}} \to 0$. Using the triangle inequality and (4.6b), we have

$$\|Ax_k - b\| \leq \|Ax_k + \delta r_k - b\| + \delta\|r_k\| \leq x_k^T z_k / \gamma_{\text{P}} + \delta\|r_k\| \quad \text{for all } k \in \mathcal{K}.$$

Theorem 5.4 ensures that $x_k^T z_k \to 0$. Our assumption that $\{r_k\}_{\mathcal{K}} \to 0$ then guarantees that $A\bar{x} = b$.

Part 2 is a direct application of Theorem 5.5 with $s_* = 0$, and part 3 is a direct consequence of parts 1–2 and Theorem 5.5. □

Note that in part 3 of Theorem 5.6 it is not sufficient to assume that both $\liminf_{k\in\mathbb{N}} \|r_k\| = 0$ and $\liminf_{k\in\mathbb{N}} \|s_k\| = 0$ occur because those lower limits may occur along disjoint subsequences. The assumption of part 3 thus has to be slightly more restrictive. However, if the algorithm generates a single limit point, then in fact that limit point must be a primal–dual solution, as the following corollary establishes.

Corollary 5.7 *Suppose Algorithm 5.1 with $\epsilon = 0$ generates the sequence $\{w_k\}$ such that $\{(x_k, y_k, z_k)\}$ has the single limit point $(\bar{x}, \bar{y}, \bar{z})$ and that there exists $\alpha_* > 0$ such that $\alpha_k \geq \alpha_*$ for all sufficiently large k. Then $(\bar{x}, \bar{y}, \bar{z})$ is a primal–dual solution of (P)–(D).*

Proof Because α_k is uniformly bounded away from zero for all sufficiently large k, $\{x_k\} \to \bar{x}$ implies that the step $\Delta x_k \to 0$, which together with (4.5b), in turn implies that

$$\|s_{k+1}\| = \|s_k + \alpha_k \Delta s_k\| \leq (1 - \alpha_k)\|s_k\| + \alpha_k\|\Delta x_k\| \leq (1 - \alpha^*)\|s_k\| + \|\Delta x_k\|.$$

Taking the limit superior in the above shows that $\{s_k\} \to 0$.

Because Δy similarly converges to zero, we have from (4.5a)

$$\|r_{k+1}\| \leq (1 - \alpha^*)\|r_k\| + \|\Delta y_k\|$$

and we obtain as above that $\{r_k\} \to 0$.

The conclusion then follows from Theorem 5.6(3). □

It it interesting to see that $\Delta x_k \to 0$ and $\alpha_k \geq \alpha^*$ suffice to establish that $\{s_k\} \to 0$ in Corollary 5.7. It seems difficult however to dispense with the assumption that $\Delta y_k \to 0$, though this is a counter-intuitive assumption in the degenerate case. This is because r and Δy are directly connected by (4.5a). Assuming only that $\Delta z_k \to 0$, we would be able to establish from (4.5c) that $\{A^T \Delta y_k\} \to 0$, which combines with (4.5a) to imply that $\{A^T r_k\} \to 0$. The latter does not imply primal feasibility of $\bar{x}$ but instead shows that $\bar{x}$ solves the linear least-squares problem

$$\text{minimize } \|b - Ax\|_2.$$

Indeed from $(Ax_k + \delta r_k - b) \to 0$, we deduce that $\{r_k\} \to (b - A\bar{x})/\delta$ and therefore that $A^T(b - A\bar{x}) = 0$.

Theorem 5.6 suggests a computational test to detect infeasible problems. Along the iterations, the evolution of $\|r_k\|$ and $\|s_k\|$ indicates whether feasibility will be achieved or whether the problem is (locally) infeasible. We describe the precise test used in our implementation in Sect. 6.

5.3 Second variation: decreasing regularization

Our second method works with variable regularization parameters ρ_k and δ_k and enforces conditions (4.6a), (4.6d) and (4.6e). It is specified as Algorithm 5.2.

Algorithm 5.2 Variation of the primal–dual method with variable regularization

Apply Algorithm 4.1 with the following specializations. In Step 3, only conditions (4.6a), (4.6d), and (4.6e) are enforced, and ρ_{k+1} and δ_{k+1} are chosen so that

$$\rho_{k+1} \geq \kappa_\rho \rho_k, \tag{5.15a}$$

$$\delta_{k+1} \geq \kappa_\delta \delta_k, \tag{5.15b}$$

for some constants $0 < \kappa_\rho < 1$ and $0 < \kappa_\delta < 1$.

Note that the statement of Algorithm 4.1 states that $\{\rho_k\}$ and $\{\delta_k\}$ are nonincreasing sequences while (5.15a) and (5.15b) require that they not decrease faster than linearly.

The convergence properties of Algorithm 5.2 are similar to those of Algorithm 5.1. We establish a result mirroring Theorem 5.5. The main difference is that ρ and δ now vary along nonincreasing sequences and that by construction $\rho_k s_k$ and $\delta_k r_k$ converge

to zero. This stronger property allows us to recover a solution of (P)–(D) provided the regularization parameters do not decrease too rapidly.

Theorem 5.8 *Suppose Algorithm 5.2 with $\epsilon = 0$ generates the sequence $\{w_k\}$ and that $\{(x_k, z_k)\}$ remains bounded. Suppose also that there exist $k_0 \in \mathbb{N}$ and $\alpha^* \in (0, 1]$ such that $\alpha_k \geq \alpha^*$ for all $k \geq k_0$. Then the sequence $\{(x_k, r_k, s_k, z_k)\}$ is bounded and every limit point of $\{(x_k, z_k)\}$ determines a primal–dual solution of the primal–dual pair (P)–(D).*

Proof Our assumption that $\alpha_k \geq \alpha^* > 0$ for all $k \geq k_0$ and (4.9) imply, as in the end of the proof of Theorem 5.4, that $x_k^T z_k \to 0$. As a consequence μ_k also converges to zero. By definition of $\mathcal{N}_k$ we also have $\|\delta_k r_k\| \leq n\mu_k/\gamma_R$ and $\|\rho_k s_k\| \leq n\mu_k/\gamma_S$, and thus $\delta_k r_k \to 0$ and $\rho_k s_k \to 0$.

We have from (4.6d), (5.7) and (5.15b) that

$$\begin{aligned}
\|Ax_{k+1} + \delta_{k+1} r_{k+1} - b\| &= \|(1-\alpha_k)(Ax_k + \delta_k r_k - b) + (\delta_{k+1} - \delta_k) r_{k+1}\| \\
&\leq (1-\alpha_k)\|Ax_k + \delta_k r_k - b\| + \frac{\delta_k - \delta_{k+1}}{\delta_{k+1}} \frac{n}{\gamma_R} \mu_{k+1} \\
&\leq (1-\alpha^*)\|Ax_k + \delta_k r_k - b\| + (1-\kappa_\delta)\frac{n}{\gamma_R}\mu_{k+1},
\end{aligned}$$

for all $k \geq k_0$, where we used the fact that $\alpha_k \geq \alpha^*$. Upon taking the limit superior in the last inequality, and using the fact that $\mu_{k+1} \to 0$, we obtain that $\{Ax_k + \delta_k r_k - b\} \to 0$.

We prove in a similar fashion that $\{c + Qx_k + \rho_k s_k - A^T y_k - z_k\} \to 0$.

In view of the above, our assumption that $\{(x_k, z_k)\}$ remains bounded then implies that $\{A^T y_k\}$ remains bounded. Let $g_k := A^T y_k$ for all k and consider any limit point $(\bar{x}, \bar{g}, \bar{z})$ of $\{(x_k, g_k, z_k)\}$. We have in the limit

$$c + Q\bar{x} - \bar{g} - \bar{z} = 0, \quad A\bar{x} = b, \quad \bar{X}\bar{z} = 0, \quad (\bar{x}, \bar{z}) \geq 0.$$

Because $g_k \in \text{Range}(A^T)$ for all k and this subspace is closed, $\bar{g}$ still lies in $\text{Range}(A^T)$, i.e., there exists $\bar{y} \in \mathbb{R}^m$ such that $\bar{g} = A^T \bar{y}$. □

5.4 Third variation: fixed regularization and the unsymmetric system

We present in this section a minor variation of Algorithm 5.1 that is based on solving the 3-by-3 block system (4.3), instead of the reduced system (4.4). The benefit of this approach is that it allows us to dispense with the counter-intuitive assumption, used throughout Sect. 5.2, that $\{z_k\}$ is bounded.

Algorithm 5.3 Variation on Algorithm 5.1: constant regularization

Apply Algorithm 4.1 with $\rho_k := \rho_0 > 0$ and $\delta_k := \delta_0 > 0$ for all k. In Step 2, Δw_k is computed from (4.3) instead of (4.4). In Step 3, only conditions (4.6a), (4.6b), and (4.6c) are enforced.

The key is the following counterpart to Lemma 5.3, which replaces the assumption on the boundedness of $\{z_k\}$ with the assumption that complementarity pairs remain bounded away from zero. This alternate assumption is used only as a device in the proof of Theorem 5.10 in order to prove a contradiction; it is not needed for the final convergence result.

Lemma 5.9 *Suppose that* $(\Delta x, \Delta y, \Delta z)$ *is given by Step 2 of Algorithm 5.3, that the sequence* $\{(r_k, s_k)\}$ *is bounded, and that there is a constant* $\bar{\mu} > 0$ *such that* $[x_k]_i[z_k]_i \geq \bar{\mu}$ *for all iterations* k *and all* $i = 1, \ldots, n$. *Then there exists a constant* π, *dependent only on* n, *such that*

$$|\Delta x_i \Delta z_i| \leq \pi, \quad |\Delta x_i \Delta z_i - \gamma_c \Delta x^T \Delta z/n| \leq \pi, \quad \text{and} \quad |\Delta x_i \Delta z_i - \bar{\gamma}_c \Delta x^T \Delta z/n| \leq \pi.$$

Proof Using the neighborhood conditions (4.6a)–(4.6c) and our assumption that $\{r_k\}$ and $\{s_k\}$ are bounded, it is easy to see that the right-hand side of (4.3) is uniformly bounded. Because $[x_k]_i[z_k]_i \geq \bar{\mu}$ for all iterations k and all $i = 1, \ldots, n$, we may invoke Armand and Benoist [4, Theorem 1] to conclude that the matrix of (4.3) is nonsingular and has a uniformly-bounded inverse. Consequently, $(\Delta x, \Delta y, \Delta z)$ also remains uniformly bounded. □

The following result is exactly the same as Theorem 5.4, except that there is no boundedness assumption on the iterates $\{z_k\}$.

Theorem 5.10 *Suppose that Algorithm 5.3 with* $\epsilon = 0$ *generates the sequence* $\{w_k\}$, *and that the sequence* $\{(r_k, s_k)\}$ *is bounded. Then* $\{\mu_k\} \to 0$.

Proof Assume by contradiction that $\mu_k \not\to 0$, i.e., there exists a constant $\bar{\mu} > 0$ such that $[x_k]_i[z_k]_i \geq \bar{\mu}$ for all iterations k and all $i = 1, \ldots, n$. Using Lemma 5.9, we may repeat the proof of Theorem 5.4 verbatim to conclude that $\mu_k \to 0$, a contradiction. □

Similarly, Theorems 5.5, 5.6 and Corollary 5.7 hold verbatim without the assumption that $\{z_k\}$ remains bounded.

Ideally, we would like to similarly relax the assumptions used in analysis of the decreasing-regularization algorithm of Sect. 5.3. However, the technique used to prove Theorem 5.8 does not allow us to relax the boundedness assumption on either $\{x_k\}$ or $\{z_k\}$, or the assumption that the steplengths are bounded away from zero. It is however worthwhile noting that—thanks to Armand and Benoist [4, Theorem 1]—it is easy to establish that $\mu_k \to 0$ by contradiction without assuming that the steplengths are bounded away from zero. It is necessary for this to modify the update of δ_k to

$$\delta_{k+1} \geq \max(\kappa_\mu \mu_k, \kappa_\delta \delta_k),$$

for some constants $\kappa_\mu > 0$ and $0 < \kappa_\delta < 1$. Assuming by contradiction that there exists $\bar{\mu} > 0$ such that $\mu_k \geq \bar{\mu}$ for all k, we immediately obtain that $\delta_k \geq \kappa_\mu \bar{\mu} > 0$ for all k. Under boundedness of $\{(x_k, z_k)\}$, the right-hand side of (4.3) remains bounded.

Hence, invoking Armand and Benoist [4, Theorem 1], we obtain $\mu_k \to 0$, a contradiction. This convergence result, although weaker than Theorem 5.8, comes with weak assumptions.

6 Implementation

We have implemented the primal–dual regularization methods described in Algorithm 4.1 (and specialized in Algorithms 5.1 and 5.2) in the Python programming language as part of the NLPy programming platform for optimization [29]. Our implementation handles convex quadratic programs in slack formulation, i.e., the primal–dual pair

$$\underset{x,t}{\text{minimize}}\ c^T x + \tfrac{1}{2} x^T Q x \quad \text{subject to}\ Ax + Bt = b,\ t \geq 0 \tag{6.1a}$$

$$\underset{x,y,z}{\text{maximize}}\ b^T y - \tfrac{1}{2} x^T Q x \quad \text{subject to}\ Qx - A^T y = c,\ B^T y + z = 0,\ z \geq 0, \tag{6.1b}$$

where $c \in \mathbb{R}^n$, $Q \in \mathbb{R}^{n \times n}$, $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{m \times p}$, $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, $t \in \mathbb{R}^p$, $y \in \mathbb{R}^m$, and $z \in \mathbb{R}^n$. An intermediate routine in NLPy is used to reformulate problems with more general upper and lower constraints into this formulation by adding slack variables as needed. In the remainder of this section we outline our main implementation choices.

6.1 Newton-step computation

The Newton systems corresponding to (6.1) that need to be solved in Step 2 of Algorithm 4.1 take the form

$$\begin{bmatrix} -(Q+\rho I) & 0 & A^T \\ 0 & -(T^{-1}Z + \rho I) & B^T \\ A & B & \delta I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta t \\ \Delta y \end{bmatrix} = \begin{bmatrix} c + Qx - A^T y \\ -B^T y - \mu T^{-1} e \\ b - Ax - Bt \end{bmatrix}.$$

As this system reveals, primal regularization plays a double role here. On one hand, it helps handle free variables transparently; this is most clearly seen when $Q = 0$. On the other hand, it regularizes the system when A, the constraint matrix associated with the free variables, does not have full column rank.

These linear systems are solved using the multifrontal symmetric indefinite factorization package MA57 of the Harwell Subroutine Library [17]. The ordering scheme used by MA57 to promote sparsity of the factors is the multilevel recursive-bisection and k-way partitioning implemented in the MeTiS package [18,19]. Because the sparsity pattern of the linear systems remains fixed throughout all iterations, only a single `analyze` phase is necessary, and the resulting ordering and elimination tree are retained until termination. Thus, only the `factorize` and `solve` phases need be performed at each execution of Step 2.

In the case where $Q = 0$, the implementation uses a rescaling of (6.2) advocated by Saunders [33]. The resulting transformed system continues to allow for a one-time symbolic factorization, and has the benefit of yielding a system with an *effective* condition number [14] that is proportional to $1/(\rho\delta)$.

Finally, our implementation recovers Δz using the last block equation from (4.2) instead of (4.5c), i.e.,

$$\Delta z = -z + \sigma\mu T^{-1}e - T^{-1}Z\Delta t.$$

Indeed the latter is cheaper to implement as it only requires vector operations.

6.2 Initial point

Our implementation adapts the initial-point procedure advocated by Mehrotra [23] for LPs to our QP formulation (6.1). The initial values for the primal variables (x, t) are chosen as the solution of

$$\underset{x,t}{\text{minimize}} \ \tfrac{1}{2}x^T Qx + \tfrac{1}{2}\|t\|^2 \quad \text{subject to } Ax + Bt = b, \tag{6.2}$$

and the initial values of the dual variables (y, z) are chosen as the solution of

$$\underset{u,y,z}{\text{minimize}} \ \tfrac{1}{2}u^T Qu + \tfrac{1}{2}\|z\|^2 \quad \text{subject to } Qu - A^T y = -c, \ B^T y + z = 0. \tag{6.3}$$

Both t and z are subsequently modified using the procedure described by Mehrotra [23] to ensure sufficient positivity. The primal–dual residuals r and (s_x, s_t) that correspond to those in (2.1)–(2.2) are then initialized so as to satisfy exactly the perturbed primal and dual feasibility conditions at the initial point, i.e.,

$$\begin{aligned} \delta_0 r &= b - Ax \\ \rho_0 s_x &= A^T y + z - c - Qx \\ \rho_0 s_t &= B^T y + z, \end{aligned}$$

where ρ_0 and δ_0 are the initial regularization parameters.

The initial-value problems (6.2) and (6.3) may be solved via the systems

$$\begin{bmatrix} -Q & 0 & A^T \\ 0 & -I & B^T \\ A & B & 0 \end{bmatrix} \begin{bmatrix} x \\ t \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} -Q & 0 & A^T \\ 0 & -I & B^T \\ A & B & 0 \end{bmatrix} \begin{bmatrix} u \\ z \\ y \end{bmatrix} = \begin{bmatrix} c \\ 0 \\ 0 \end{bmatrix};$$

a single factorization suffices to solve both systems. Because the above coefficient matrix is precisely that used in the iterations of Algorithm 4.1, and to account for the cases where Q is singular or $[A \ B]$ is rank deficient, we regularize the $(1, 1)$ and $(3, 3)$ blocks, i.e., we change the $(1, 1)$ block to $-(Q + \rho I)$ and the $(3, 3)$ block to δI with $\rho = \delta = 10^{-4}$. This permits an efficient implementation in which the analyze phase of the factorization is performed a single time while computing the initial guess, and subsequent iterations only need to recompute the numerical factorization.

6.3 Long-step and predictor-corrector variants

In this section and the next, we use the formulation (P)–(D) instead of (6.1) as this simplifies the exposition considerably.

Our implementation deviates from Algorithm 4.1 in the method for computing the steplength of Step 3, and does not force the iterates to remain in the neighborhood $\mathcal{N}_k$ defined by (4.6). Instead, only the so-called *fraction-to-the-boundary* rule is applied. The largest steplengths to the boundary of the nonnegative orthant are computed from the formulae

$$\alpha_x^{\max} = \min_{[\Delta x]_i < 0} \left\{ 1, -\frac{x_i}{[\Delta x]_i} \right\} \quad \text{and} \quad \alpha_z^{\max} = \min_{[\Delta z]_i < 0} \left\{ 1, -\frac{z_i}{[\Delta z]_i} \right\}, \tag{6.4}$$

and the final steplengths are selected as

$$\alpha_x = \tau \alpha_x^{\max} \quad \text{and} \quad \alpha_z = \tau \alpha_z^{\max},$$

where the safeguard $\tau \in (0, 1)$. In our implementation, we set

$$\tau = \max\{0.995, 1 - \mu\}.$$

The centering parameter is selected at each iteration as $\sigma = \min\{0.1, 100\mu\}$.

At the user's option, the long-step algorithm may enforce an equal steplength $\alpha = \min\{\alpha_x, \alpha_z\}$ for all variables, or follow Kojima et al. [20] and use α_x to update the primal variables x and r; and α_z to update the dual variables s, y and z; the latter option is the default in our implementation.

A predictor-corrector variation on Algorithm 4.1 was also implemented as an option. It follows Mehrotra [23], except that the linear systems are primal–dual regularized and in augmented form. The variation is outlined in Algorithm 6.1.

Mehrotra's stepsize selection heuristic is used in Step 3 of Algorithm 6.1. If $\Delta x \geq 0$, the same rule as in the long-step method is applied. Otherwise, let i_{P} be the index realizing the minimum in the computation of $\alpha_x^{\max}$ in (6.4). The factor γ_{P} is now computed so as to satisfy

$$(x_{i_{\text{P}}} + \gamma_{\text{P}} \alpha_x^{\max} [\Delta x]_{i_{\text{P}}})(z_{i_{\text{P}}} + \alpha_z^{\max} [\Delta z]_{i_{\text{P}}}) = \tfrac{1}{100} \mu_+,$$

where

$$\mu_+ := (x + \alpha_x^{\max} \Delta x)^T (z + \alpha_z^{\max} \Delta z)/n.$$

Finally, the primal stepsize is set to

$$\alpha_x = \max\{1 - \tfrac{1}{100}, \gamma_{\text{P}}\} \alpha_x^{\max}.$$

Algorithm 6.1 Variation of the primal–dual method with predictor-corrector steps

Apply Algorithm 4.1, where Step 2 is replaced by the following steps.

Step 2a. [Step computation: predictor] Compute a predictor step $\Delta w^{\mathrm{P}} = (\Delta x^{\mathrm{P}}, \Delta r^{\mathrm{P}}, \Delta s^{\mathrm{P}}, \Delta y^{\mathrm{P}}, \Delta z^{\mathrm{P}})$ from w_k using (4.4), with $\sigma = 0$, i.e.,

$$\begin{bmatrix} -(Q + X_k^{-1} Z_k + \rho_k I) & A^T \\ A & \delta_k I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} c + Qx_k - A^T y_k \\ b - Ax_k \end{bmatrix},$$

and (4.5). Compute

$$\begin{aligned} \alpha_x^{\mathrm{P}} &= \arg\max \left\{ \alpha \in (0, 1] \mid x_k + \alpha \Delta x^{\mathrm{P}} \geq 0 \right\}, \\ \alpha_z^{\mathrm{P}} &= \arg\max \left\{ \alpha \in (0, 1] \mid z_k + \alpha \Delta z^{\mathrm{P}} \geq 0 \right\}, \\ \mu^{\mathrm{P}} &= (x_k + \alpha_x^{\mathrm{P}} \Delta x^{\mathrm{P}})^T (z_k + \alpha_z^{\mathrm{P}} \Delta z^{\mathrm{P}})/n. \end{aligned}$$

Step 2b. [Step computation: corrector] Set $\sigma = (\mu^{\mathrm{P}}/\mu_k)^3$ and compute a combined predictor-centering-corrector step $(\Delta x^k, \Delta y^k)$ as the solution to

$$\begin{bmatrix} -(Q + X_k^{-1} Z_k + \rho_k I) & A^T \\ A & \delta_k I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} c + Qx_k - A^T y_k - \sigma \mu_k X_k^{-1} e - \Delta X^{\mathrm{P}} \Delta z^{\mathrm{P}} \\ b - Ax_k \end{bmatrix},$$

where $\Delta X^{\mathrm{P}} = \mathrm{diag}(\Delta x^{\mathrm{P}})$, and recover $(\Delta r^k, \Delta s^k, \Delta z^k)$ from (4.5).

The dual stepsize α_z is computed similarly unless $\Delta z \geq 0$. Note that if i_{D} is the index realizing the minimum in the computation of $\alpha_z^{\max}$ in (6.4), the calculation of γ_{P} is not defined when $i_{\mathrm{P}} = i_{\mathrm{D}}$. In this case again, we fall back on the same stepsize rule as in the long-step method. When there are several choices for i_{P} and i_{D}, the largest index is retained.

7 Numerical experiments

We tested Algorithm 6.1 on LPs from the Netlib [26] collection, including the so-called Kennington problems, and on a set of convex QPs drawn from Maros and Mészáros [22], CUTEr [16], and the Brunel [25] collections, all in the AMPL modeling language. The latter test set provides variations of the Netlib LPs with a quadratic objective function. For each problem, we select $\rho_0 = \delta_0 = 1$, and use the update rules

$$\rho_{k+1} = \frac{\rho_k}{10}, \qquad \delta_{k+1} = \frac{\delta_k}{10}.$$

Safeguards are used to ensure that ρ_k and δ_k never take values less than 10^{-8}. Thus, our implementation may be viewed as a predictor-corrector variant of Algorithm 5.2 that branches into Algorithm 5.1 once the regularization parameters have attained their minimum allowed value. These update rules are meant to satisfy (5.15) while allowing the regularization parameters to decrease sufficiently fast on non-degenerate problems.

Our implementation of Algorithm 6.1 declares successful termination as soon as

$$\frac{\|c + Qx - A^T y - z\|}{1 + \|c\|_2 + \|A\|_F + \|Q\|_F} \leq \epsilon,$$
$$\frac{\|b - Ax\|}{1 + \|b\|_2 + \|A\|_F + \|Q\|_F} \leq \epsilon,$$
$$\frac{\mu}{1 + \|c^T x\|_2 + \|A\|_F + \|Q\|_F} \leq \epsilon,$$

for a user-defined tolerance $\epsilon > 0$, where the third condition is a measure of the duality gap between (P) and (D). Primal (local) infeasibility is declared if

$$\mu_k < 10^{-2}\epsilon\mu_0 \quad \text{and} \quad \|\rho_k s_k\| > \frac{1}{10^6\epsilon} \min_{0\leq\ell\leq k} \|\rho_\ell s_\ell\|$$

occur for three consecutive iterations. Similarly, dual (local) infeasibility is declared if

$$\mu_k < 10^{-2}\epsilon\mu_0 \quad \text{and} \quad \|\delta_k r_k\| > \frac{1}{10^6\epsilon} \min_{0\leq\ell\leq k} \|\delta_\ell r_\ell\|$$

occur for six consecutive iterations.

We factorize the Newton systems with MA57 by setting the pivot threshold parameter to 10^{-15}. In all of our tests, this has always resulted in a Cholesky-like factorization and no 2×2 pivots were ever considered by the factorization routine. In some cases, A is rank deficient and either or both of ρ_k and δ_k are too small to sufficiently regularize the system. In this case they are both increased by a factor of 100. If this happens for five consecutive iterations, we declare inability to sufficiently regularize the system and abort.

We perform comparisons with the state-of-the-art implementations of the predictor-corrector method PCx [8] for linear programming, and OOQP [12] for convex quadratic programming. PCx solves a system of the form (1.2) at each iteration using the Cholesky factorization of Ng and Peyton [27] while paying special attention to (nearly) dense columns in A. OOQP works with the augmented system (1.1) and uses MA57 to perform a symmetric indefinite factorization (with 2×2 pivots). The above stopping conditions are comparable to those of PCx and OOQP.

7.1 Results on linear programs

PCx was run with a single corrector step in order to imitate the standard predictor-corrector method used by our implementation. All stopping tolerances were set to 10^{-6} and presolve was turned off in order for all codes to use the same set of preprocessing primitives provided by AMPL. Scaling and iterative refinement options were enabled.

Table 1 reports results of applying our implementation of Algorithm 6.1 to the feasible problems from the Netlib collection; problem `forplan` was excluded because PCx reports an error while reading the AMPL data file translated from the original

Table 1 Results on the feasible test set from the Netlib collection

Name	It	Obj	Pfeas	Dfeas	Gap	Setup	Solve	Stat	PCx
25fv47	21	5.50489466e+03	1.1e−04	2.1e−06	9.4e−04	0.01	0.56	opt	
80bau3b	37	9.89157958e+05	8.5e−05	1.0e−03	5.3e−03	0.06	3.87	opt	7
adlittle	11	2.25496885e+05	1.6e−07	3.8e−05	1.7e−05	0.00	0.02	opt	
afiro	10	−4.64753143e+02	4.0e−12	2.8e−10	3.1e−10	0.00	0.01	opt	
agg	28	−3.59917644e+07	9.7e−06	4.5e−05	6.3e−06	0.00	0.08	opt	
agg2	22	−2.02391606e+07	1.0e−02	1.4e−04	1.8e−05	0.01	0.12	opt	
agg3	22	1.03121565e+07	1.6e−04	3.6e−05	1.4e−05	0.00	0.13	opt	
bandm	15	−1.58622591e+02	6.8e−06	2.0e−07	5.1e−05	0.00	0.08	opt	
beaconfd	11	3.35960979e+04	8.7e−07	7.3e−06	1.7e−04	0.00	0.03	opt	
blend	12	−3.08120451e+01	7.8e−07	3.3e−06	1.1e−05	0.00	0.02	opt	
bnl1	42	1.97777158e+03	2.2e−04	7.2e−07	3.5e−04	0.01	0.60	opt	
bnl2	25	1.81489644e+03	9.2e−04	7.4e−05	3.9e−03	0.03	1.11	opt	2
boeing1	17	−3.35160799e+02	4.0e−05	5.6e−07	3.3e−04	0.00	0.14	opt	1
boeing2	17	−3.15016202e+02	7.5e−06	3.0e−07	4.7e−05	0.00	0.05	opt	2
bore3d	15	1.37308381e+03	1.6e−07	6.2e−07	2.9e−05	0.00	0.04	opt	11
brandy	17	1.51856654e+03	7.9e−05	6.8e−06	9.6e−05	0.00	0.07	opt	1?
capri	19	2.33221872e+03	2.5e−07	1.4e−07	1.4e−04	0.00	0.11	opt	
cycle	29	−5.21690750e+00	2.7e−06	1.1e−06	2.5e−04	0.02	1.30	opt	1
czprob	27	2.18563265e+06	4.0e−03	1.7e−06	3.3e−04	0.02	0.78	opt	
d2q06c	22	1.22957389e+05	3.6e−03	5.3e−05	2.6e−03	0.04	2.34	opt	3
d6cube	15	3.15492127e+02	5.1e−07	3.5e−06	1.8e−06	0.04	4.91	opt	1
degen2	12	−1.43517637e+03	3.1e−05	5.5e−09	1.8e−06	0.01	0.14	opt	10
degen3	13	−9.87292210e+02	1.2e−05	2.0e−07	2.4e−06	0.03	0.94	opt	12
dfl001	44	1.12927661e+07	2.0e−04	4.0e−07	3.2e−03	0.08	44.37	opt	72?
e226	16	−1.87508664e+01	4.1e−07	2.6e−08	1.1e−04	0.00	0.09	opt	
etamacro	20	−7.55672949e+02	1.3e−07	2.9e−07	3.3e−04	0.01	0.17	opt	
fffff800	27	5.55682732e+05	1.1e−03	2.0e−05	6.5e−06	0.01	0.44	opt	
finnis	17	1.72808427e+05	1.5e−04	1.6e−05	3.7e−04	0.00	0.14	opt	
fit1d	14	−9.14601140e+03	1.9e−05	7.1e−06	4.4e−05	0.01	0.66	opt	
fit1p	35	9.15255524e+03	2.9e−06	2.0e−07	1.8e−03	0.01	0.89	opt	
fit2d	17	−6.81095623e+04	3.0e−04	6.7e−08	6.7e−03	0.06	47.50	opt	
fit2p	27	6.86829801e+04	1.2e−04	1.1e−04	1.0e−02	0.09	8.20	opt	
ganges	16	−1.09584202e+05	6.4e−02	3.8e−07	2.5e−05	0.01	0.29	opt	
gfrd-pnc	12	6.90633908e+06	7.1e−03	1.2e−03	1.0e−03	0.01	0.15	opt	
greenbea	44	−7.24583904e+07	6.9e−03	1.9e−04	8.5e−05	0.04	2.91	opt	8?
greenbeb	25	−4.30122857e+06	5.8e−03	9.4e−05	4.7e−04	0.03	1.86	opt	6?
grow15	27	−1.06870903e+08	7.8e−03	1.3e−05	2.3e−05	0.00	0.35	opt	
grow22	28	−1.60834206e+08	2.2e−02	3.4e−05	1.5e−05	0.01	0.56	opt	
grow7	21	−4.77878109e+07	6.9e−04	3.2e−06	1.7e−08	0.00	0.13	opt	
israel	13	−8.96639842e+05	5.0e−07	1.3e−05	4.7e−05	0.00	0.05	opt	
kb2	18	−1.74989072e+03	6.6e−06	6.5e−07	1.7e−05	0.00	0.02	opt	

Table 1 continued

Name	It	Obj	Pfeas	Dfeas	Gap	Setup	Solve	Stat	PCx
lotfi	19	−2.52645539e+01	2.8e−07	1.3e−07	2.7e−05	0.00	0.08	opt	
maros-r7	28	1.50529264e+06	7.8e−08	3.6e−05	9.3e−03	0.08	18.66	opt	
maros	21	−5.80164163e+04	6.7e−03	1.1e−04	1.1e−03	0.01	0.47	opt	
modszk1	25	3.20703415e+02	4.6e−08	2.6e−08	5.8e−04	0.02	0.42	opt	1
nesm	19	1.40807614e+07	2.8e−04	1.7e−04	2.1e−03	0.01	0.71	opt	
perold	56	−9.37803725e+03	1.5e−04	6.7e−05	9.0e−04	0.01	1.46	opt	
pilot.ja	50	−6.11090894e+03	4.2e−05	8.3e−05	2.6e−03	0.01	1.61	opt	
pilot	39	−5.57313016e+02	1.0e−03	1.1e−05	5.3e−04	0.04	5.13	opt	
pilot.we	33	−2.71603413e+06	2.5e−04	5.3e−04	2.8e−03	0.01	1.10	opt	
pilot4	−315	−7.54497504e+03	2.4e−07	5.5e−04	4.8e−01	0.01	4.85	pInf	i
pilot87	31	3.02509026e+02	3.7e−06	2.8e−05	4.7e−03	0.05	10.19	opt	
pilotnov	19	−4.49694917e+03	3.0e−02	8.5e−06	1.5e−03	0.02	0.64	opt	
recipe	9	−2.66615984e+02	2.6e−06	1.0e−07	2.2e−07	0.00	0.03	opt	
sc105	16	−5.22020611e+01	5.0e−08	5.4e−09	2.2e−07	0.00	0.03	opt	
sc205	29	−5.22020543e+01	1.9e−06	1.3e−07	2.1e−05	0.01	0.09	opt	
sc50a	14	−6.45750770e+01	4.2e−12	1.0e−09	9.5e−09	0.00	0.02	opt	
sc50b	15	−7.00000000e+01	3.5e−13	2.7e−09	3.4e−08	0.00	0.02	opt	
scagr25	17	−1.47532386e+07	3.0e−04	2.0e−04	3.6e−05	0.00	0.11	opt	
scagr7	12	−2.33134744e+06	3.4e−08	6.3e−06	5.4e−05	0.00	0.03	opt	
scfxm1	16	1.84183577e+04	5.1e−05	2.7e−06	2.3e−04	0.00	0.12	opt	?
scfxm2	18	3.66642808e+04	1.7e−03	1.1e−05	2.6e−04	0.01	0.23	opt	
scfxm3	18	5.49052630e+04	4.9e−04	1.9e−05	2.6e−04	0.01	0.35	opt	
scorpion	15	1.87812502e+03	1.8e−07	1.3e−05	7.4e−07	0.00	0.07	opt	12
scrs8	20	9.04879102e+02	6.2e−06	4.6e−06	7.4e−04	0.01	0.23	opt	
scsd1	11	8.66685163e+00	9.1e−07	1.8e−06	2.2e−05	0.00	0.08	opt	
scsd6	11	5.05146566e+01	1.1e−07	1.3e−05	3.3e−04	0.01	0.13	opt	
scsd8	11	9.05000001e+02	8.6e−10	8.0e−09	1.6e−09	0.02	0.23	opt	
sctap1	18	1.41230855e+03	4.9e−07	6.9e−07	1.9e−04	0.00	0.11	opt	
sctap2	12	1.72485280e+03	1.1e−07	1.9e−06	7.0e−05	0.02	0.25	opt	
sctap3	13	1.42481706e+03	4.5e−05	1.6e−06	6.7e−04	0.02	0.35	opt	
seba	15	1.57116942e+04	3.7e−08	4.0e−05	2.5e−04	0.01	0.19	opt	
share1b	24	−7.65857416e+04	1.4e−04	2.3e−05	1.7e−04	0.00	0.09	opt	
share2b	22	−4.15729251e+02	7.3e−10	9.2e−08	1.3e−05	0.00	0.05	opt	
shell	16	1.20899593e+09	6.0e−02	3.5e−03	1.9e−04	0.01	0.21	opt	11
ship04l	12	1.79332470e+06	2.6e−07	2.2e−08	1.1e−07	0.01	0.20	opt	1
ship04s	12	1.79872152e+06	2.2e−05	2.2e−06	4.4e−06	0.01	0.13	opt	
ship08l	13	1.90906481e+06	1.1e−05	4.4e−03	1.2e−05	0.02	0.37	opt	
ship08s	12	1.92014172e+06	5.4e−05	3.4e−03	3.0e−05	0.01	0.17	opt	
ship12l	16	1.47019198e+06	3.9e−05	1.0e−08	2.8e−06	0.02	0.58	opt	
ship12s	15	1.48923694e+06	7.0e−06	8.7e−06	7.1e−07	0.01	0.26	opt	
sierra	13	1.53998710e+07	1.9e−04	3.4e−04	1.6e−03	0.01	0.42	opt	19

Table 1 continued

Name	It	Obj	Pfeas	Dfeas	Gap	Setup	Solve	Stat	PCx
stair	33	−2.51249662e+02	2.5e−04	9.2e−06	1.9e−04	0.00	0.23	opt	
standata	11	1.25769954e+03	1.4e−09	5.3e−08	1.4e−07	0.01	0.13	opt	
standgub	11	1.25769954e+03	1.4e−09	5.3e−08	1.4e−07	0.01	0.13	opt	
standmps	20	1.40704994e+03	1.2e−05	6.1e−06	9.1e−04	0.01	0.24	opt	
stocfor1	17	−4.11319717e+04	1.3e−06	1.2e−07	1.3e−07	0.00	0.04	opt	
stocfor2	21	−3.90191967e+04	7.4e−05	6.0e−07	3.1e−04	0.02	0.65	opt	
truss	19	4.58824902e+05	2.6e−07	1.3e−04	2.5e−05	0.05	1.52	opt	
tuff	19	2.92165655e−01	6.9e−04	5.2e−07	2.7e−03	0.00	0.21	opt	
vtp.base	22	1.24324258e+05	7.9e−05	6.9e−07	4.9e−06	0.00	0.06	opt	
wood1p	23	1.44263639e+00	2.8e−06	1.4e−10	1.1e−04	0.02	3.45	opt	12
woodw	20	1.30821113e+00	3.5e−05	2.9e−10	2.0e−03	0.03	1.61	opt	

MPS format. The columns are, from left to right, the problem name, number of iterations, final objective value, the (unscaled) primal feasibility, (unscaled) dual feasibility, (unscaled) duality gap, setup time, solve time, and exit status; the last column gives information on the exit status of PCx applied to that problem. The setup time includes the time to scale the constraint matrix A. In the 'Stat' column, 'opt' signals that an optimal solution was found, 'pInf' indicates that the problem is categorized as primal infeasible, and 'dInf' indicates dual infeasibility.

The last column of Table 1 gives a statistic about PCx's performance on the problem: a number indicates the iteration at which PCx first encountered a rank-deficient Newton system that had to be modified. On those problems, a naive implementation of the interior-point method would fail without recourse to an ad-hoc modification of the Newton system. The flag 'i' indicates that PCx reported the problem to be infeasible and the flag '?' indicates termination with unknown status.

Table 2 reports results on the Kennington problems.

Figure 1 compares the number of iterations with that of PCx using performance profiles. Note that there is a performance cost for systematically regularizing but that the payoff is in robustness. This is especially apparent on the feasible problems. Only problem `pilot4` was incorrectly classified as primal infeasible. Note also from the scale of the horizontal axis in the rightmost plot that PCx is consistently, but only marginally, more efficient.

7.2 Results on convex quadratic programs

The AMPL interface of OOQP does not allow the user to select a single corrector step per iteration. Instead, the method chooses to perform zero, one, or several corrector steps based on local information. Similarly, the stopping tolerance cannot be set by the user and is preset to 10^{-8}. The comparisons with OOQP below should thus be interpreted in the light of these discrepancies. They remain meaningful in our view because the dominant cost of both algorithms is the symmetric indefinite factorization and a single such factorization is performed per iteration. Indeed, performing multiple

Table 2 Results on the Kennington test set from the Netlib collection

Name	It	Obj	Pfeas	Dfeas	Gap	Setup	Solve	Stat	PCx
cre-a	21	2.36860031e+07	5.2e−04	1.2e−06	6.4e−03	0.03	1.28	opt	22
cre-b	39	2.37519301e+07	4.9e−04	1.0e−02	3.1e−02	0.17	23.21	opt	42
cre-c	24	2.53605569e+07	4.9e−04	7.7e−03	4.5e−03	0.03	1.13	opt	26
cre-d	43	2.50493568e+07	5.6e−04	2.1e−03	2.8e−02	0.14	19.73	opt	40
ken-07	14	−6.79520013e+08	2.1e−04	1.9e−04	2.1e−06	0.02	0.52	opt	13
ken-11	20	−6.97229838e+09	2.8e−03	1.7e−03	3.0e−05	0.11	6.66	opt	21
ken-13	21	−1.02567896e+10	2.3e−04	2.8e−03	2.1e−04	0.22	17.87	opt	23
ken-18	26	−5.21787063e+10	3.7e−02	4.1e−03	1.7e−03	0.77	96.43	opt	30
osa-07	40	5.35722642e+05	2.4e−05	8.4e−07	3.3e−07	0.14	61.22	opt	1
osa-14	40	1.10646608e+06	5.2e−04	4.0e−03	1.7e−05	0.25	270.62	opt	
osa-30	39	2.14210935e+06	1.0e−03	3.8e−05	9.9e−06	0.46	923.52	opt	1
osa-60	47	4.04406978e+06	1.4e−05	3.2e−06	5.5e−07	1.05	5,594.74	opt	2
pds-02	14	2.88583769e+10	1.2e−03	9.4e−02	6.8e−05	0.04	1.13	opt	24
pds-06	21	2.77626335e+10	6.8e−02	5.0e−03	1.5e−04	0.14	11.88	opt	35
pds-10	27	2.67294538e+10	6.9e−02	1.7e−03	2.3e−04	0.33	38.07	opt	35
pds-20	37	2.38262235e+10	6.0e−02	3.0e−04	5.2e−04	0.52	264.49	opt	43

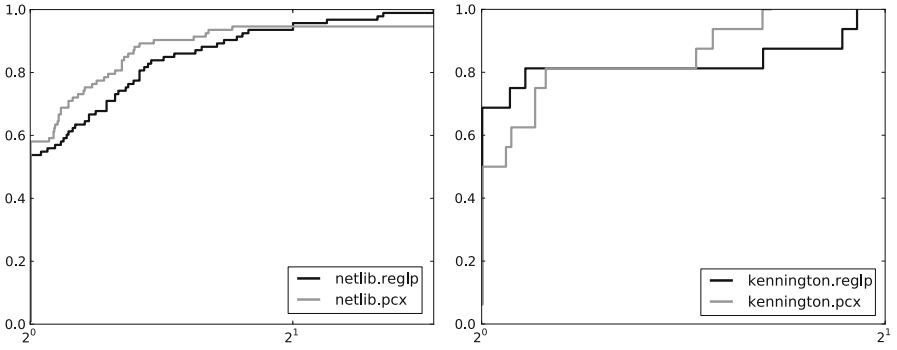

Fig. 1 Iteration profiles comparing our implementation with PCx on linear programs from the Netlib collection. Note the differing horizontal-axis scales

corrections amounts to performing multiple system solves with the same matrix. Table 3 reports results on the Brunel problems. The columns are as in the previous section. Values in the last column thus indicate problems on which a naive implementation of the interior-point method would fail because of degeneracy. Finally, a '?' indicates termination with unknown status.

Similarly, Tables 4 and 5 report results on a set of convex QPs from the CUTEr and Maros and Meszaros collections, respectively.

Figure 2 summarizes performance in terms of number of iterations in comparison with OOQP. As before, we see improved robustness with very comparable

Table 3 Results on the Brunel test set

Name	It	Obj	Pfeas	Dfeas	Gap	Setup	Solve	Stat	OOQP
q25fv47	37	1.37444479e+07	8.3e−10	1.8e−05	5.5e−08	0.23	1.69	opt	
qadlittl	11	4.80318859e+05	1.7e−08	7.3e−07	7.0e−11	0.00	0.02	opt	
qafiro	10	−1.59078179e+00	2.3e−12	1.9e−11	1.9e−11	0.00	0.01	opt	
qbandm	16	1.63523445e+04	2.2e−05	9.1e−05	1.5e−04	0.00	0.09	opt	
qbeaconf	19	1.64712060e+05	1.8e−08	3.1e−05	4.5e−10	0.00	0.08	opt	
qbore3d	15	1.30771532e+03	8.7e−05	1.8e−04	1.1e−05	0.00	0.06	opt	17
qbrandy	27	2.83751156e+04	7.0e−07	2.4e−05	5.2e−07	0.00	0.14	opt	19
qcapri	45	5.50432154e+07	6.4e−08	3.5e−05	2.4e−04	0.00	0.31	opt	
qe226	15	2.12653444e+02	5.5e−09	4.0e−05	9.4e−08	0.00	0.08	opt	
qetamacr	19	3.13324640e+03	1.4e−09	1.3e−03	3.0e−09	0.01	0.33	opt	
qffff80	38	8.73147461e+05	2.5e−08	1.2e−04	1.3e−08	0.01	0.59	opt	?
qforplan	39	3.89586283e+09	1.0e−08	4.7e−06	5.2e−04	0.00	0.28	opt	
qgfrdxpn	37	1.00790563e+11	8.5e−05	4.2e−02	2.9e−03	0.01	0.46	opt	6
qgrow15	31	−1.01693640e+08	5.3e−08	5.8e−07	1.0e−12	0.00	0.36	opt	
qgrow22	30	−1.49628953e+08	3.1e−08	4.6e−05	3.0e−14	0.01	0.52	opt	
qgrow7	26	−4.27987139e+07	5.7e−08	7.9e−07	7.3e−12	0.00	0.15	opt	
qisrael	17	2.53478377e+07	1.4e−05	2.0e−04	5.2e−07	0.00	0.06	opt	
qpilotno	38	1.93996045e+06	1.1e−02	1.0e−08	2.7e+01	0.01	1.24	opt	
qrecipe	−18	−2.11843700e+09	1.5e−07	1.5e+00	1.0e+00	0.00	0.05	dInf	
qsc205	10	−5.81382400e−03	4.3e−11	2.6e−10	1.7e−09	0.00	0.03	opt	
qscagr25	18	2.01737938e+08	1.6e−10	1.8e−09	1.4e−12	0.00	0.11	opt	
qscagr7	15	2.68659498e+07	2.9e−05	3.9e−04	3.9e−06	0.00	0.03	opt	
qscfxm1	34	1.68826916e+07	8.2e−10	2.0e−06	6.1e−09	0.00	0.22	opt	11
qscfxm2	41	2.77761616e+07	1.2e−08	1.3e−04	6.6e−07	0.01	0.57	opt	14
qscfxm3	41	3.08163545e+07	1.9e−09	1.8e−06	1.4e−08	0.01	0.86	opt	15
qscorpio	16	1.88050955e+03	2.2e−09	6.8e−09	5.1e−10	0.01	0.07	opt	17?
qscrs8	21	9.04582297e+02	7.3e−07	2.7e−03	2.5e−05	0.01	0.21	opt	
qscsd1	10	8.66780789e+00	9.8e−06	9.9e−05	6.1e−06	0.00	0.06	opt	
qscsd6	13	5.08082727e+01	3.6e−08	9.0e−05	2.6e−07	0.01	0.14	opt	
qscsd8	14	9.40763574e+02	1.8e−10	1.7e−11	1.5e−09	0.02	0.30	opt	
qsctap1	21	1.41586146e+03	7.6e−08	1.5e−04	5.7e−07	0.00	0.12	opt	
qsctap2	13	1.73502650e+03	2.0e−09	7.7e−08	3.6e−09	0.01	0.27	opt	
qsctap3	14	1.43880937e+03	9.5e−08	2.5e−04	5.0e−05	0.02	0.40	opt	1
qseba	29	8.14818004e+07	3.8e−09	1.2e−04	1.5e−07	0.01	0.37	opt	
qshare1b	23	7.20078319e+05	5.0e−10	3.0e−06	2.6e−07	0.00	0.08	opt	
qshare2b	22	1.17036955e+04	6.3e−09	5.7e−05	1.5e−05	0.00	0.04	opt	
qshell	32	1.15485056e+12	2.1e−10	1.9e−08	1.9e−12	0.11	0.83	opt	22
qship04l	14	2.42001687e+06	4.9e−06	7.1e−04	9.9e−07	0.01	0.29	opt	16
qship04s	14	2.42499391e+06	1.7e−06	4.7e−04	3.4e−07	0.01	0.21	opt	17
qship08l	15	2.37604062e+06	1.3e−08	3.2e−04	1.1e−09	0.13	0.98	opt	16
qship08s	14	2.38572893e+06	1.3e−06	7.3e−04	5.5e−08	0.04	0.40	opt	20
qship12l	18	3.01887658e+06	1.6e−08	5.0e−04	3.8e−09	0.30	1.80	opt	20
qship12s	17	3.05696227e+06	6.2e−08	1.8e−03	3.1e−08	0.05	0.59	opt	18
qsierra	19	2.27507369e+07	5.7e−05	4.8e−02	4.7e−06	0.01	0.53	opt	20
qstair	31	2.63637390e+06	8.0e−08	6.7e−08	2.2e−04	0.01	0.24	opt	?
qstandat	10	3.85050337e−08	2.3e−08	8.0e−07	2.3e−10	0.01	0.10	opt	

Table 4 Results on the CUTEr test set

Name	It	Obj	Pfeas	Dfeas	Gap	Setup	Solve	Stat	OOQP
aug2d	7	1.68741165e+06	2.5e−06	1.1e−07	5.0e−06	0.16	0.69	opt	1
aug2dc	7	1.81836794e+06	3.0e−06	1.3e−07	4.2e−06	0.16	0.69	opt	
aug2dcqp	15	6.49814687e+06	2.0e−07	3.1e−08	2.6e−03	0.16	3.23	opt	
aug2dqp	15	6.23701204e+06	3.0e−09	3.9e−09	7.7e−08	0.16	3.20	opt	
aug3d	5	5.54067726e+02	7.4e−09	2.3e−08	4.3e−10	0.03	0.09	opt	1
aug3dc	5	7.71262435e+02	7.1e−08	6.8e−08	2.0e−09	0.02	0.07	opt	
aug3dcqp	10	9.93362229e+02	2.4e−10	5.2e−05	1.3e−07	0.02	0.34	opt	
aug3dqp	9	6.75245193e+02	2.0e−09	3.0e−09	4.8e−06	0.02	0.29	opt	
cvxqp1_l	180	1.07185846e+08	8.9e−03	1.1e−08	6.7e−01	0.22	241.85	opt	2
cvxqp1_m	21	1.08750764e+06	4.5e−06	9.2e−11	6.2e−05	0.01	0.63	opt	1
cvxqp1_s	12	1.15907181e+04	2.2e−09	1.7e−06	5.1e−10	0.00	0.02	opt	
cvxqp2_l	33	8.18424578e+07	1.7e−06	8.0e−02	4.6e−07	0.21	24.08	opt	
cvxqp2_m	20	8.20155428e+05	7.3e−07	5.5e−03	2.9e−08	0.01	0.41	opt	
cvxqp2_s	13	8.12094048e+03	7.3e−09	9.0e−04	1.4e−07	0.00	0.03	opt	
cvxqp3_l	−261	1.12474427e+08	1.4e−02	3.3e+03	1.4e+00	0.24	407.60	dInf	
cvxqp3_m	25	1.31590642e+06	3.5e−03	8.1e−09	7.0e−01	0.01	0.88	opt	
cvxqp3_s	14	1.19434322e+04	4.7e−09	1.7e−05	3.6e−09	0.00	0.03	opt	
dtoc3	1	0.00000000e+00	0.0e+00	0.0e+00	0.0e+00	0.13	0.00	opt	
dual1	7	3.50125357e−02	3.1e−05	3.1e−04	7.1e−10	0.01	0.02	opt	
dual2	5	3.37341266e−02	1.1e−06	5.5e−07	4.8e−09	0.01	0.01	opt	
dual3	7	1.35755886e−01	7.9e−07	3.6e−05	1.9e−11	0.01	0.02	opt	
dual4	6	7.46092921e−01	2.1e−05	5.9e−04	5.0e−09	0.00	0.01	opt	
dualc1	17	6.15234907e+03	1.1e−06	6.2e−10	4.6e−07	0.00	0.04	opt	
dualc2	14	3.55120405e+03	1.9e−06	1.3e−02	9.6e−08	0.00	0.03	opt	
dualc5	7	4.27184602e+02	1.6e−04	1.8e−08	4.6e−07	0.00	0.02	opt	
dualc8	13	1.83079215e+04	4.2e−05	6.8e−02	6.2e−08	0.00	0.05	opt	
genhs28	4	9.27162518e−01	6.6e−06	2.1e−06	4.0e−07	0.00	0.00	opt	
gouldqp2	6	2.12809706e−04	1.8e−05	4.8e−05	9.9e−05	0.00	0.05	opt	
gouldqp3	11	2.06278461e+00	6.4e−11	1.3e−04	2.1e−07	0.01	0.09	opt	
hs118	13	6.64821428e+02	6.5e−10	2.9e−06	1.0e−05	0.00	0.01	opt	
hs21	7	−9.99600000e+01	7.1e−12	5.9e−10	2.1e−09	0.00	0.00	opt	
hs21mod	7	−9.59599732e+01	1.2e−08	8.0e−08	6.4e−06	0.00	0.01	opt	
hs268	3	2.38098936e−02	6.0e−03	5.1e−02	1.8e−06	0.00	0.00	opt	
hs35	5	1.11111156e−01	9.2e−08	2.7e−07	3.3e−09	0.00	0.00	opt	
hs35mod	5	1.11111144e−01	7.0e−08	2.0e−07	2.9e−09	0.00	0.00	opt	
hs51	5	1.60493840e−12	5.3e−07	6.1e−08	2.1e−09	0.00	0.00	opt	
hs52	3	5.32486697e+00	1.3e−04	2.6e−05	6.0e−06	0.00	0.00	opt	
hs53	4	4.09318054e+00	1.9e−05	1.5e−05	1.5e−05	0.00	0.00	opt	
hs76	5	−4.68181818e+00	1.9e−08	1.8e−08	2.3e−09	0.00	0.00	opt	
hues-mod	13	3.48244639e+07	1.7e−07	4.6e−07	2.6e−04	0.05	0.90	opt	
huestis	17	3.48244639e+11	3.2e−07	1.2e−08	1.4e−03	0.06	1.05	opt	

Table 4 continued

Name	It	Obj	Pfeas	Dfeas	Gap	Setup	Solve	Stat	OOQP
ksip	11	5.75797939e−01	3.7e−09	2.2e−05	2.2e−07	0.01	0.15	opt	
liswet1	26	2.50162200e+01	5.4e−06	1.5e−10	3.3e−06	0.10	2.26	opt	
liswet10	23	2.50024218e+01	2.5e−06	8.1e−05	3.2e−07	0.10	1.90	opt	
liswet11	26	2.50016718e+01	9.1e−06	1.7e−04	9.5e−06	0.10	2.37	opt	
liswet12	−45	2.51125653e+01	3.4e−05	4.6e−03	5.2e−04	0.10	4.75	pInf	
liswet2	24	2.49979481e+01	2.4e−07	1.3e−11	1.1e−08	0.10	1.97	opt	
liswet3	28	2.50012115e+01	6.2e−08	2.2e−04	3.5e−09	0.10	2.31	opt	
liswet4	24	2.50000803e+01	2.3e−07	1.8e−04	9.0e−09	0.10	1.96	opt	
liswet5	24	2.50342340e+01	1.9e−07	2.6e−04	3.9e−09	0.10	1.98	opt	
liswet6	24	2.49957073e+01	2.8e−07	1.6e−04	4.1e−09	0.10	1.97	opt	
liswet7	24	2.50027670e+01	3.5e−06	1.3e−11	1.1e−06	0.10	1.95	opt	
liswet8	26	2.50027231e+01	9.9e−06	1.9e−04	1.1e−05	0.10	2.33	opt	
liswet9	35	2.50068585e+01	3.9e−05	1.5e−04	4.1e−04	0.10	3.73	opt	
lotschd	9	2.39841589e+03	3.4e−11	1.3e−11	2.9e−10	0.00	0.01	opt	
mosarqp1	8	−9.52875441e+02	1.9e−08	1.4e−05	1.8e−09	0.02	0.16	opt	
mosarqp2	8	−1.59748195e+03	1.9e−08	2.4e−09	6.7e−08	0.01	0.08	opt	
powell20	29	5.20895751e+10	4.5e−05	4.9e−08	2.8e+01	0.10	2.23	opt	1
primal1	5	−3.50122553e−02	2.8e−07	6.4e−05	1.8e−08	0.00	0.03	opt	
primal2	4	−3.37335358e−02	1.0e−06	7.2e−06	9.5e−10	0.00	0.03	opt	
primal3	5	−1.35755784e−01	5.9e−09	1.0e−06	3.8e−10	0.01	0.08	opt	
primal4	17	−7.46080929e−01	1.1e−11	4.0e−05	1.8e−07	0.01	0.21	opt	
primalc1	18	−6.15525083e+03	3.7e−08	1.0e−07	1.5e−07	0.00	0.05	opt	
primalc2	16	−3.55130760e+03	5.5e−09	2.0e−09	1.7e−08	0.00	0.04	opt	
primalc5	12	−4.27229166e+02	1.5e−08	3.1e−04	9.5e−07	0.00	0.04	opt	
primalc8	14	−1.83094299e+04	4.1e−09	1.7e−08	2.2e−09	0.00	0.07	opt	
qpcblend	16	−7.84245304e−03	7.1e−09	4.6e−05	1.8e−09	0.00	0.03	opt	
qpcboei1	31	1.15039140e+07	6.5e−09	1.6e−09	2.0e−05	0.00	0.23	opt	?
qpcboei2	60	7.43625764e+06	7.3e−03	8.7e−08	7.3e+02	0.00	0.21	opt	7?
qpcstair	27	7.17440738e+05	1.2e−08	6.2e−10	1.6e−05	0.01	0.17	opt	
s268	3	2.38098936e−02	6.0e−03	5.1e−02	1.8e−06	0.00	0.00	opt	
stcqp1	12	1.55143551e+05	1.5e−06	3.5e−04	7.9e−07	0.09	0.87	opt	11
stcqp2	10	2.23273133e+04	5.5e−07	8.3e−04	7.3e−09	0.10	0.85	opt	
tame	5	0.00000000e+00	8.3e−10	2.2e−09	2.8e−10	0.00	0.00	opt	
ubh1	5	0.00000000e+00	3.2e−08	1.3e−08	4.8e−09	0.12	0.74	opt	
yao	6	2.80158414e−03	6.1e−06	9.2e−05	3.1e−06	0.02	0.08	opt	
zecevic2	5	−4.12500000e+00	1.4e−08	8.2e−09	1.0e−10	0.00	0.00	opt	

performance. Problems `qreceipe`, `cvxqp3_l`, `liswet12`, and `stadat1` are miscategorized as infeasible. In the cases of `liswet12` and `stadat1`, the final residuals are sufficiently small to suggest that this is due to our particular implementation of the infeasibility detection.

Table 5 Results on the Maros and Meszaros test set

Name	It	Obj	Pfeas	Dfeas	Gap	Setup	Solve	Stat	OOQP
boyd1	31	−6.17352214e+07	1.4e−04	9.7e−01	5.7e−07	1.16	31.09	opt	
boyd2	106	2.12578364e+01	3.5e−09	4.4e−05	4.1e−05	1.15	229.07	opt	
cont-050	9	−4.56384168e+00	2.7e−08	9.6e−05	1.0e−04	0.03	0.43	opt	
cont-100	13	−4.64439804e+00	2.0e−08	4.9e−08	5.4e−08	0.09	3.52	opt	
cont-101	11	1.95523371e−01	1.3e−07	9.6e−06	3.8e−06	0.09	3.30	opt	
cont-200	13	−4.68487802e+00	1.2e−07	2.8e−04	3.1e−04	0.41	19.46	opt	
cont-201	16	1.92481206e−01	1.2e−07	2.8e−07	1.8e−07	0.35	29.91	opt	
cont-300	13	2.05463886e−01	3.5e−07	1.5e−06	2.8e−05	0.81	69.27	opt	
dpklo1	4	3.68371210e−01	6.3e−04	7.5e−05	2.5e−06	0.00	0.01	opt	
exdata	20	−1.41842405e+02	1.4e−09	9.4e−05	4.1e−06	68.86	16.40	opt	
laser	19	2.40960135e+06	8.2e−10	7.3e−11	1.9e−07	0.02	0.21	opt	
qptest	6	4.37187495e+00	6.0e−09	1.3e−09	2.2e−09	0.00	0.00	opt	
stadat1	−691	−2.86599806e+07	9.6e−04	5.0e−09	5.0e−04	0.04	31.06	dInf	7?
stadat2	47	−3.26271739e+01	2.3e−06	5.0e−05	2.4e−05	0.05	1.72	opt	
stadat3	58	−3.57857770e+01	1.2e−05	1.7e−04	6.2e−05	0.07	4.42	opt	
values	10	−1.39662082e+00	1.4e−10	8.6e−06	9.7e−09	0.01	0.03	opt	

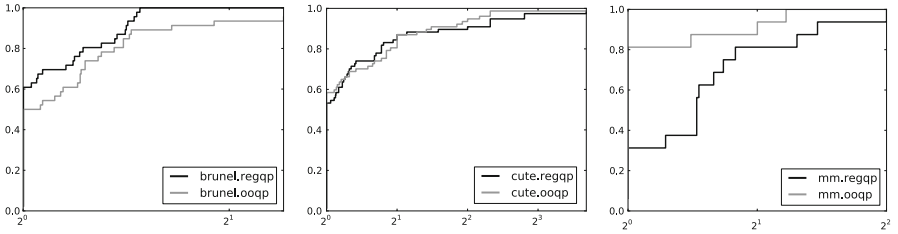

Fig. 2 Iteration profiles comparing our implementation with OOQP on convex quadratic programs from the Brunel, CUTEr and Maros and Meszaros collection. Recall that OOQP performs multiple corrector steps per predictor step

8 Looking ahead

Anjos and Burer [3] provide a convergence proof for the primal regularization scheme proposed by Mészáros [24] in the context of second-order cone programming:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad c^T x \quad \text{subject to } Ax = b,\ x \in \mathcal{K},$$

where $\mathcal{K}$ is the symmetric self–dual cone. Both Mészáros [24] and Anjos and Burer [3] motivate this type of regularization as a means to handle free variables. The convergence proof of Anjos and Burer [3] is similar to ours but is weaker in at least three respects. The first is that it only considers primal regularization. The second is that it explicitly assumes boundedness of the Newton direction—an assumption done away with in the present paper. The third is that it assumes that A has full row rank.

Yet more importantly, their regularization parameter update occurs posthumously in the sense that the parameter value is adjusted if the Newton direction just computed violates some condition. A clear disadvantage is then that each time the parameter value is changed, the Newton direction must be recomputed.

Mixed linear complementarity problems (mLCPs) have the form

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} z \\ 0 \end{bmatrix} = \begin{bmatrix} c \\ b \end{bmatrix}, \quad (x, z) \geq 0, \quad Xz = 0,$$

where the matrix is square and positive semi-definite (but not necessarily symmetric), and can be shown to be formally equivalent to the convex quadratic programming problem [43, Theorem 8.4]. In particular, this implies that both M_{11} and M_{22} are positive semi-definite. Primal–dual regularized direction-finding systems for the mLCP will take the form

$$\begin{bmatrix} -(M_{11} + X^{-1}Z + \rho I) & M_{12} \\ M_{21} & M_{22} + \delta I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} M_{11}x + M_{12}y + c - \sigma\mu X^{-1}e \\ b - M_{21}x - M_{22}y \end{bmatrix}.$$

Because of the relationships between the class of mLCPs and the class of convex QPs, it is reasonable to expect that all results above apply directly to problems formulated as mLCPs. However, a direct interpretation in terms of primal–dual regularized problems such as (2.1) and (2.2) is lost, as are the symmetry and quasi-definiteness of the coefficient matrices. The parameters ρ and δ nevertheless help treat problems for which M_{11} is either singular or ill conditioned and problems for which either or both of M_{12} and M_{21} is rank deficient.

The strong similarity between interior-point methods for convex QP and for larger classes of problems, such as second-order cone problems and semidefinite programming problems, also suggests that our results carry over to those more general classes.

There are theoretical advantages to basing an interior-point algorithm on the 3-by-3 block system (4.3) instead of the reduced system (4.4); the assumptions for convergence are weaker and the proofs are simpler. It remains to be seen whether or not the 3-by-3 block system also has advantages in the practice.

Acknowledgments The research of Prof. M.P. Friedlander was supported in part by a Discovery grant from the Natural Sciences and Engineering Council of Canada. The research of Prof. D. Orban was supported in part by NSERC Discovery grant 299010-04.

References

1. Altman, A., Gondzio, J.: Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization. Optim. Methods Softw. **11**(12), 275–302 (1999)
2. Altman, A., Gondzio, J. Higher order primal dual method (2009). http://www.maths.ed.ac.uk/~gondzio/software/hopdm.html
3. Anjos, M.F., Burer, S.: On handling free variables in interior-point methods for conic linear optimization. SIAM J. Optim. **18**(4), 1310–1325 (2007)
4. Armand, P., Benoist, J.: Uniform boundedness of the inverse of a jacobian matrix arising in regularized interior-point methods. Math. Program. (2011). doi:10.1007/s10107-011-0498-3

5. Bellavia, S., Gondzio, J., Morini, B.: Regularization and preconditioning of KKT systems arising in nonnegative least-squares problems. Numer. Linear Algebra Appl. **16**(1), 39–61 (2009). doi:10.1002/nla.610
6. Bunch, J.R., Parlett, B.N.: Direct methods for solving symmetric indefinite systems of linear equations. SIAM J. Numer. Anal. **8**(4), 639–655 (1971)
7. Castro, J., Cuesta, J.: Quadratic regularizations in an interior-point method for primal block-angular problems. Math. Programm., 1–31 (2010). doi:10.1007/s10107-010-0341-2
8. Czyzyk, J., Mehrotra, S., Wagner, M., Wright, S.J.. PCx user guide version 1.1. Technical Report OTC 96/01, Optimization Technology Center, Evanston (1996). http://www.mcs.anl.gov/OTC/Tools/PCx
9. Fletcher, R.: Practical Methods of Optimization, 2nd edn. Wiley, Chichester (1987)
10. Friedlander, M.P., Leyffer, S.: Global and finite termination of a two-phase augmented Lagrangian filter method for general quadratic programs. SIAM J. Sci. Comput. **30**(4), 1706–1729 (2008). doi:10.1137/060669930
11. Friedlander, M.P., Tseng, P.: Exact regularization of convex programs. SIAM J. Optim. **18**(4), 1326–1350 (2007). doi:10.1137/060675320
12. Gertz, E.M., Wright, S.J.: Object-oriented software for quadratic programming. ACM Trans. Math. Softw. **29**(1), 58–81 (2003)
13. Gill, P.E., Murray, W., Ponceleón, D.B., Saunders, M.A.: Solving reduced KKT systems in barrier methods for linear and quadratic programming. Technical Report SOL 91-7, Systems Optimization Laboratory, Stanford University, Stanford (1991)
14. Gill, P.E., Saunders, M.A., Shinnerl, J.R.: On the stability of Cholesky factorization for symmetric quasidefinite systems. SIAM J. Matrix Anal. Appl. **17**(1), 35–46 (1996)
15. Gondzio, J.: Matrix-free interior point method. Comput. Optim. Appl., 1–24 (2011). doi:10.1007/s10589-010-9361-3
16. Gould, N.I.M., Orban, D., Toint, P.L.: CUTEr and SifDec, a Constrained and Unconstrained Testing Environment, revisited. ACM Trans. Math. Softw. **29**(4), 373–394 (2003)
17. Harwell Subroutine Library: A collection of Fortran codes for large-scale scientific computation. AERE Harwell Laboratory (2000). http://www.numerical.rl.ac.uk/hsl
18. Karypis, G., Kumar, V.: Multilevel k-way partitioning scheme for irregular graphs. J. Parallel Distrib. Comput. **48**(1), 96–129 (1998)
19. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. **20**(1), 359–392 (1999)
20. Kojima, M., Megiddo, N., Mizuno, S.: A primal–dual infeasible-interior-point algorithm for linear programming. Math. Program. **61**, 263–280 (1993)
21. Mangasarian, O.L., Meyer, R.R.: Nonlinear perturbation of linear programs. SIAM J. Control Optim. **17**(6), 745–752 (1979)
22. Maros, I., Mészáros, C.: A repository of convex quadratic programming problems. Optim. Methods Softw. **11, 12**, 671–681 (1999) (Special Issue on Interior Point Methods)
23. Mehrotra, S.: On the implementation of a primal–dual interior-point method. SIAM J. Optim. **2**(4), 575–601 (1992)
24. Mészáros, C.: On free variables in interior point methods. Optim. Methods Softw. **9**, 121–139 (1998)
25. Mittelmann, H.: http://plato.la.asu.edu/ftp/ampl_files/qpdataampl (2006)
26. Netlib. http://netlib.org/lp (2011)
27. Ng, E.G., Peyton, B.W.: Block sparse cholesky algorithms on advanced uniprocessor computers. SIAM J. Sci. Comput. **14**(5), 1034–1056 (1993). doi:10.1137/0914063. http://link.aip.org/link/?SCE/14/1034/1
28. Oliveira, A.R.L., Sorensen, D.C.: A new class of preconditioners for large-scale linear systems from interior point methods for linear programming. Linear Algebra Appl. **394**, 1–24 (2005)
29. Orban, D.: NLPy—a large-scale optimization toolkit in Python. Technical Report Cahier du GERAD G-2010-xx, GERAD, Montréal (2010). http://nlpy.sourceforge.net/.
30. Rockafellar, R.T.: The multiplier method of Hestenes and Powell applied to convex programming. J. Optim. Theory Appl. **12**, 555–562 (1973)
31. Rockafellar, R.T.: Augmented Lagrangians and applications of the proximal point algorithm in convex programming. Math. Oper. Res. **1**(2), 97–116 (1976)
32. Rusten, T., Winther, R.: A preconditioned iterative method for saddlepoint problems. SIAM J. Matrix Anal. Appl. **13**(3), 887–904 (1992). doi:10.1137/0613054. http://link.aip.org/link/?SML/13/887/1

33. Saunders, M.A.: Solution of sparse rectangular systems using LSQR and CRAIG. BIT **35**, 588–604 (1995)
34. Saunders, M.A. : Cholesky-based methods for sparse least squares: The benefits of regularization. In: Adams, L., Nazareth, J.L. (eds.) Linear and Nonlinear Conjugate Gradient-Related Methods., pp. 92–100. SIAM, Philadelphia (1996)
35. Saunders, M.A.: PDCO: Primal–dual interior method for convex objectives (2010). http://www.stanford.edu/group/SOL/software/pdco.html
36. Saunders, M.A., Tomlin, J.A.: Solving regularized linear programs using barrier methods and KKT systems. SOL Report 96-4, Dept. of EESOR, Stanford University (1996)
37. Setiono, R.: Interior proximal point algorithm for linear programs. J. Optim. Theory Appl. **74**(3), 425–444 (1992)
38. Setiono, R.: Interior dual proximal point algorithm for linear programs. Eur. J. Oper. Res. **77**, 96–110 (1994)
39. Silvester, D., Wathen, A.: Fast iterative solution of stabilised Stokes systems part II: using general block preconditioners. SIAM J. Numer. Anal. **31**(5), 1352–1367 (1994). doi:10.1137/0731070. http://link.aip.org/link/?SNA/31/1352/1
40. Vanderbei, R.J.: Interior-point methods: algorithms and formulations.. ORSA J. Comput. **6**(1), 32–34 (1994)
41. Vanderbei, R.J.: Symmetric quasi-definite matrices. SIAM J. Optim. **5**(1), 100–113 (1995)
42. Wright, S.J.: Implementing proximal point methods for linear programming. J. Optim. Theory Appl. **65**((3), 531–554 (1990)
43. Wright, S.J.: Primal–Dual Interior-Point Methods. SIAM, Philadelphia (1997)