

Addressing Rank Degeneracy in Constraint-Reduced Interior-Point Methods for Linear Optimization

Luke B. Winternitz* André L. Tits[†] P.-A. Absil[‡]

November 13, 2012

Abstract

In earlier work (Tits *et al.*, SIAM J. Optim., 17(1):119–146, 2006; Winternitz *et al.*, COAP, 51(3):1001–1036, 2012), the present authors and their collaborators proposed primal-dual interior-point (PDIP) algorithms for linear optimization that, at each iteration, use only a subset of the (dual) inequality constraints in constructing the search direction. For problems with many more variables than constraints in primal form, this can yield a major speedup in the computation of search directions. However, in order for the Newton-like PDIP steps to be well defined, it is necessary that the gradients of the constraints included in the working set span the full dual space. In practice, in particular in the case of highly sparse problems, this often results in an undesirably large working set—or in an expensive trial-and-error process for its selection. In this paper we present two approaches that remove this non-degeneracy requirement, while retaining the convergence results obtained in the earlier work.

Key words: linear programming linear optimization constraint reduction primal-dual interior-point regularization

AMS Subject Classification: 90C51; 90C05; 90C06; 90C34; 65K05

1 Introduction

Primal-dual interior-point (PDIP) methods for the solution of linear optimization problems (LP) apply Newton’s method, or variations thereof, to the equality portion of the perturbed Karush-Kuhn-Tucker (KKT) optimality conditions for the primal-dual pair; see, e.g., [Wri97] for details. In this paper, we focus on “imbalanced” problems, i.e., problems in which, in standard form, the constraint matrix has many more columns than rows. These are problems with many more variables than constraints in their standard primal form, or equivalently many more (inequality) constraints than variables in their standard dual form (2). Specifically, following [TAW06] and [WNT012], we consider “constraint-reduced” interior-point algorithms for solving such problems. The main idea of this class of algorithms is that at each iteration, a small working set of constraints is chosen and a PDIP-type direction is computed for the resulting “reduced” primal-dual pair

*NASA - Goddard Space Flight Center, Mail Stop 596, Greenbelt, MD 20771, USA, (Luke.B.Winternitz@nasa.gov)

[†]Corresponding author. Department of Electrical and Computer Engineering & Institute for Systems Research, University of Maryland College Park, MD 20742 USA, (andre@umd.edu). The work of this author was supported by DOE grant DESC0002218.

[‡]Department of Mathematical Engineering, ICTEAM Institute, Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium, (absil@inma.ucl.ac.be)

by solving the “reduced” Newton-KKT system. While the algorithms analyzed in [TAW06] and [WNT012] require that an initial strictly dual-feasible vector be provided, in [HT12], an exact-penalty-function framework with automatic penalty adjustment is proposed and analyzed, within which these algorithms are readily extended to handle arbitrary initial points. In [TAW06, WNT012, HT12], global convergence and local quadratic convergence of the proposed algorithms is proved under mild assumptions. Numerical tests reported in those papers and in [Win10], both on randomly generated problems and on problems arising from specific engineering applications, demonstrate the power of constraint reduction on imbalanced problems; further, a comparison with the “revised primal simplex with partial pricing” of [BT97], on filter design applications, is reported in Chapter 5 of [Win10] (see in particular Table 5.4 in that Ph.D. dissertation) that shows that constraint reduction is competitive with simplex on certain such imbalanced problems and superior on others.

The Newton-KKT linear system specifies a well-defined search direction (i.e., the matrix associated with that system is non-singular) if and only if the “reduced” constraint matrix has full row-rank. Simple and intuitive rules for selecting the constraints to use at each iteration do not necessarily guarantee that the rank of this matrix will be full. Thus, any constraint-reduced algorithm must eventually deal with such possible “rank-degeneracy”, which is the term we use to refer to the occurrence of working set of constraints which results in rank-deficient reduced constraint matrix. Such occurrence, of course, is possible even when the un-reduced constraint matrix has full row-rank, a condition that is required by most interior-point methods and can be enforced by preprocessing and, for example, holds in the problem shown in Figure 1.

In earlier work on constraint-reduced algorithms, such as [DY91] and [TAW06], rank-degeneracy is precluded by assuming that every full-dimension square sub-matrix of the constraint matrix is non-singular and then simply requiring that the cardinality size of the working set always be at least equal to the number of rows of the constraint matrix. However, such assumption is unlikely to hold in general, in real world problems, and furthermore, may be impossible to verify *a priori*. In other prior work, including that of [WNT012], the assumption is imposed on the working set rather than on the constraint matrix. That is, the constraint selection rule requires that the former be chosen so that the corresponding reduced constraint matrix have full rank. It is possible to explicitly enforce this, for example, with a Gram-Schmidt orthogonalization procedure, by updating a rank-revealing pivoted Cholesky factorization, or with simple schemes that add more constraints to the working set when the full-rank condition fails. However, these methods may be computationally expensive, and may result in large working sets, which is contrary to the general intent of these algorithms.

In this paper, we investigate ways to efficiently deal with rank-degeneracy. Specifically, we first propose to *regularize* the Newton-KKT system, as was previously done, for example, in [ST96] and [FO12].¹ A second proposed algorithm, the “Kernel-Step” algorithm, can be viewed as a limit of the regularized algorithm as the regularization parameter goes to zero. Both algorithms are analyzed and proved to converge globally and locally with quadratic rate under mild assumptions. Effectiveness of both algorithms is demonstrated by numerical experimentation, which suggest that, depending on the class of problems of interest, one or the other of these algorithms is superior.

The remainder of this paper is organized as follows. Section 2 establishes the context and the notation.

¹A preliminary version of the latter was the object of a conference presentation in 2007 [FO07], which contributed inspiration for the PhD research of the first author of the present paper (see [Win10]), from which the present paper is issued.

Section 3 reviews key ideas from [ST96] on the regularization of linear systems such as (9). The Regularized rPDAS and Kernel-Step rPDAS algorithms are stated and analyzed in section 4, and numerical results are reported in section 5. Section 6 is devoted to concluding remarks.

2 Preliminaries

We consider the primal and dual standard forms of linear optimization:

$$\begin{aligned} \min c^T x \\ \text{s.t. } Ax = b, \\ x \geq 0, \end{aligned} \quad \text{and} \quad \begin{aligned} \max b^T y \\ \text{s.t. } A^T y \leq c, \end{aligned} \quad (1)$$

where A , an $m \times n$ matrix, is “imbalanced”, i.e., $n \gg m$, that is, the dual problem has many more inequality constraints than variables. We assume $b \neq 0$. (This assumption is benign, since if $b = 0$ the problem at hand is readily solved: any dual-feasible point y^0 —assumed available for the algorithms analyzed here—is dual optimal and—under our dual feasibility assumption— $x = 0$ is primal optimal.) The dual problem can alternatively be written in the form (with slack variable s)

$$\begin{aligned} \max b^T y \\ \text{s.t. } A^T y + s = c, \\ s \geq 0. \end{aligned} \quad (2)$$

As mentioned above, PDIP methods apply Newton’s method, or variations thereof, to the equality portion of the perturbed Karush-Kuhn-Tucker (KKT) optimality conditions for the primal-dual pair (1), namely, of

$$\begin{aligned} A^T y + s - c &= 0, \\ Ax - b &= 0, \\ Xs - \tau e &= 0, \\ (x, s) &\geq 0, \end{aligned} \quad (3)$$

with $X := \text{diag}(x)$, $S := \text{diag}(s)$, e is the vector of all ones, and τ is a positive parameter. Specifically, the search direction $(\Delta x, \Delta y, \Delta s)$ solves the Newton-KKT system

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} c - A^T y - s \\ b - Ax \\ \sigma \mu e - Xs \end{pmatrix}, \quad (4)$$

where, as is typical, we have restricted τ to be a product of the form $\sigma \mu$, where $\mu := x^T s / n$, the current “duality measure”, and σ lies in $[0, 1]$. Under the usual assumption that A has full row rank, this Newton direction is well defined when X and S are positive definite, i.e., when $x > 0$ and $s > 0$, which interior-point methods enforce via a line search. The choice $\sigma := 0$ yields the “primal-dual affine-scaling” (PDAS) direction, which aims directly for the solution.

System (4) is often solved by first eliminating Δs , giving the symmetric-indefinite “KKT” or “augmented” system

$$\begin{pmatrix} -X^{-1}S & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} c - A^T y - \sigma \mu X^{-1} e \\ b - Ax \end{pmatrix}, \quad (5)$$

$$\Delta s = -A^T \Delta y + (c - A^T y - s),$$

then possibly further eliminating Δx , giving the “normal system”

$$\begin{aligned} AS^{-1}XA^T\Delta y &= b - Ax + AS^{-1}X(c - A^T y - \sigma \mu X^{-1} e), \\ \Delta s &= -A^T \Delta y + (c - A^T y - s), \\ \Delta x &= -S^{-1}X(-A^T \Delta y + c - A^T y - \sigma \mu X^{-1} e). \end{aligned} \quad (6)$$

In the dual-feasible ($s = c - A^T y > 0$) affine-scaling ($\sigma = 0$) variant, which is the primary focus of this paper, the normal equations simplify to

$$\begin{aligned} AS^{-1}XA^T\Delta y &= b, \\ \Delta s &= -A^T \Delta y, \\ \Delta x &= -x - S^{-1}X\Delta s. \end{aligned} \quad (7)$$

Let $\mathbf{n} := \{1, 2, \dots, n\}$ and let $Q \subseteq \mathbf{n}$. Throughout this paper, we use A_Q to denote the $m \times |Q|$ matrix (where $|Q|$ is the cardinality of Q) obtained from A by deleting all columns a_i with $i \notin Q$; similarly, we use x_Q and s_Q to denote the vectors of size $|Q|$ obtained from x and s by deleting all entries x_i and s_i with $i \notin Q$. In the case of “constraint-reduced” PDIP algorithms, at each iteration, starting from a strictly dual-feasible point (x, y, s) , a small working set of constraints Q is chosen and a PDIP-type direction is computed for the “reduced” primal-dual pair

$$\begin{aligned} \min c_Q^T x_Q & & \max b^T y \\ \text{s.t. } A_Q x_Q = b, & \quad \text{and} \quad & \text{s.t. } A_Q^T y + s_Q = c_Q, \\ x_Q \geq 0, & & s_Q \geq 0, \end{aligned} \quad (8)$$

by solving the “reduced” Newton-KKT system

$$\begin{pmatrix} 0 & A_Q^T & I_Q \\ A_Q & 0 & 0 \\ S_Q & 0 & X_Q \end{pmatrix} \begin{pmatrix} \Delta x_Q \\ \Delta y \\ \Delta s_Q \end{pmatrix} = \begin{pmatrix} 0 \\ b - A_Q x_Q \\ \sigma \mu_Q e - X_Q s_Q \end{pmatrix}, \quad (9)$$

with $\mu_Q := x_Q^T s_Q / |Q|$, or the block-Gauss-elimination-reduced systems analogous to (5) and (6) (or (7) when $\sigma = 0$ and $s = c - A^T y > 0$).

We use $\|\cdot\|$ to denote the 2-norm or its induced operator norm. Given a vector $x \in \mathbb{R}^n$, we let the corresponding capital letter X denote the diagonal $n \times n$ matrix with x on its main diagonal. Given a matrix M , $\mathcal{N}(M)$ and $\mathcal{R}(M)$ denote its kernel (nullspace) and range spaces. For a vector v , $[v]_-$ is defined

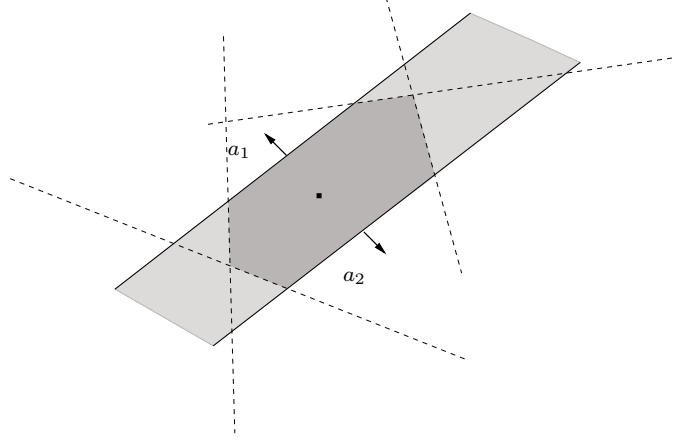


Figure 1: Example of rank-degeneracy in two dimensions (dual variable of dimension 2). The dashed and solid lines indicate constraint boundaries $\{y \mid a_i^T y = c_i\}$ and the darkly shaded area is the feasible set. From the current iterate (the dot), the two nearest constraints are selected as the working constraints. Their gradients $\{a_1, a_2\}$ do not span \mathbb{R}^2 , and hence the 2×2 reduced constraint matrix is rank deficient.

component-wise by $([v]_-)_i := \min\{v_i, 0\}$. Lowercase k always indicates an iteration count, and limits of the form $y^k \rightarrow y^*$ are meant as $k \rightarrow \infty$. Given an infinite index set K , notation such as $y^k \xrightarrow{K} y^*$ is synonymous with $y^k \rightarrow y^*$ as $k \rightarrow \infty$, $k \in K$. Further, we define the *dual-feasible*, *strictly dual-feasible*, and *dual solution sets*, respectively, as

$$\begin{aligned} F &:= \{y \in \mathbb{R}^m \mid A^T y \leq c\}, \\ F^o &:= \{y \in \mathbb{R}^m \mid A^T y < c\}, \\ F^* &:= \{y \in F \mid b^T y \geq b^T w \text{ for all } w \in F\}. \end{aligned}$$

We term a vector $y \in \mathbb{R}^m$ *stationary* if $y \in F^s$, where

$$F^s := \{y \in F \mid \exists x \in \mathbb{R}^n \text{ s.t. } Ax = b, X(c - A^T y) = 0\}. \quad (10)$$

Given $y \in F^s$, every x satisfying the conditions of (10) is referred to in this paper as *a multiplier associated to stationary point y*. A stationary vector y belongs to F^* if and only if $x \geq 0$ for some multiplier x . The *active set* at $y \in F$ is defined as

$$I(y) := \{i \in \mathbf{n} \mid a_i^T y = c_i\}.$$

3 Regularization of the linear systems

For notational simplicity, let $\sigma := 0$ (affine scaling variant). One way to control the condition number of the Newton-KKT systems, without assumption on A , is to replace system (9) (with $\sigma = 0$) with the regularized

system

$$\begin{pmatrix} 0 & A_Q^T & I_Q \\ A_Q & \delta I & 0 \\ S_Q & 0 & X_Q \end{pmatrix} \begin{pmatrix} \Delta x_Q \\ \Delta y \\ \Delta s_Q \end{pmatrix} = \begin{pmatrix} 0 \\ b - A_Q x_Q \\ -X_Q s_Q \end{pmatrix}, \quad (11)$$

where δ is a positive scalar. Such regularization was considered in [ST96] and recently in [FO12]. The corresponding augmented system is

$$\begin{pmatrix} -X^{-1}S & A^T \\ A & \delta I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} c - A^T y \\ b - Ax \end{pmatrix}. \quad (12)$$

The associated normal equations have normal matrix $A(X^{-1}S + \varrho I)^{-1}A^T + \delta I$. If $(x, s) > 0$, then $\delta > 0$ ensures that the normal matrix is non-singular.

One way to motivate the regularization (12) is by way of a “proximal-point” technique. This is achieved via the quadratic program

$$\begin{aligned} \max & b^T y - \frac{\delta}{2} \|y - y^k\|^2 \\ \text{s.t.} & A^T y \leq c. \end{aligned} \quad (13)$$

It is this approach that we will use to motivate our first algorithm, as it lets the δ -regularization in (12) enter in a simple and clean way. From this point-of-view, when the regularization parameters are set to fixed positive values, we, in effect, solve the perturbed problem (13). One can then ask what relevance the solution to this problem has the original unperturbed problem. An important result in that regard is that, under mild conditions, with a fixed value of $\delta > 0$, convergence still occurs to the solution of the *original* problem, i.e., the regularization is *exact* [FO12, Theorem 5.8]. If assumptions are made that ensure eventual solvability of the linear systems, e.g., $\text{rank}(A_Q) = m$ holds near the solution, then it may be reasonable to use such regularization in early iterations, but allow δ and ϱ to go to zero as the algorithm progresses. This is of particular interest in the context of constraint-reduction where the problem may be well-posed, but rank-degeneracy is introduced by ignoring constraints.

In the following sections, details of the analysis are often left out when they follow step by step the analysis in [TAW06] and [WNT012]. An early version of the present work can be found in the first author’s Ph.D. Dissertation [Win10].

4 Proposed Algorithms

In this section, we describe and analyze two dual-feasible constraint-reduced PDIP algorithms for linear optimization designed to deal with the possibility that $\text{rank}(A_Q) < m$. In its k th iteration, the first algorithm selects a value δ^k for the regularization parameter and computes the Newton-KKT direction for the perturbed problem (13). The second algorithm essentially takes the rPDAS step from [TAW06] whenever $b \in \mathcal{R}(A_Q)$, which of course is always the case if $\text{rank}(A_Q) = m$, but uses a different update, which we call a “kernel” step, whenever b has a component in the nullspace of A_Q^T . (A preliminary version of this second algorithm, with limited convergence analysis, was presented at ISMP06 [TAO06].) As discussed below in section 4.2.4, it turns out that the second algorithm can be naturally motivated as a limiting case of the first, for vanishing δ .

4.1 Standing assumption and Q -selection rule

Most of our global convergence analysis relies on the sole assumption of primal feasibility and dual strict feasibility. (A stronger regularity assumption, which subsumes the present one, is used in the local analysis.)

Assumption 1. The primal feasible set $\{x \mid Ax = b, x \geq 0\}$ and the dual strictly feasible set F^0 are nonempty.

In particular, our global convergence analysis does not rely on A having full row rank, nor on the primal *strictly* feasible set $\{x \mid Ax = b, x > 0\}$ being nonempty. Of course, a rank-defective A makes the problem unstable, as does emptiness of the primal strictly feasible set, and both are likely to cause difficulties in practice.

In selecting the constraint set Q , both algorithms use relaxed versions of the rule developed in [WNT012]. Below, given $\epsilon \in (0, \infty]$ and $M \in \mathbf{n}$, a set of M *most active* and the set of ϵ -*active* constraints refer, respectively, to the set of constraints with the M smallest slack values (ties broken arbitrarily) and the set of all constraints with slack value no larger than ϵ .

Definition 4.1. Let $y \in F$. Let $\epsilon \in (0, \infty]$, and let $M \in \mathbf{n}$. Then a set $Q \subseteq \mathbf{n}$ belongs to $\mathcal{Q}_{\epsilon, M}(y)$ if and only if it contains (as a subset) all ϵ -active constraints at y among some set of M most-active constraints.

Rule 4.1. At a dual-feasible point y , for $\epsilon \in (0, \infty]$ and an upper bound $M \in \{m, m + 1, \dots, n\}$ on the number of constraints active at any dual-feasible point, select Q from the set $\mathcal{Q}_{\epsilon, M}(y)$, with $|Q| > m$.

An immediate property of this rule is that all constraints that are active at the limit of a sequence $\{y^k\}$ are eventually included in Q . The condition $|Q| > m$ is used in the local convergence analysis of section 4.2.3 to ensure, under the additional assumption made there, that at least one inactive constraint is eventually included in Q .

Lemma 4.1. Let $\{y^k\}$ be a given dual-feasible sequence and let i be such that $s_i^k \rightarrow 0$ as $k \rightarrow \infty$, with $s^k = c - A^T y^k$. Further, let Q^k be selected from $\mathcal{Q}_{\epsilon, M}(y^k)$ according to Rule 4.1. Then $i \in Q^k$ for all k large enough.

Rule 4.1 differs from that in [WNT012] in that here A_Q is not required to have full rank. A secondary difference is that here $|Q|$ must be at least $m + 1$ (rather than m); this is of no practical consequence since numerical results in [WNT012] show that for good performance $|Q|$ should be greater than m .

Rule 4.1 leaves much freedom in choosing the reduced constraint set. In practice, we have had most success with specific rules that keep a small number, typically $2m$ or $3m$, most-active constraints and then add other constraints based on heuristics suggested by prior knowledge of the problem structure. When no other constraints are added, we call this the “Most-Active” rule.

4.2 Regularized rPDAS

4.2.1 Algorithm details

Now we describe and analyze our first algorithm, the “Regularized rPDAS” (Algorithm 1), which is based on the regularization (12). That is, we use constraint-reduced primal-dual *affine-scaling* (rPDAS) type steps, i.e., with $\sigma = 0$ in (4). (We briefly discuss the adaptive use of positive σ parameters below in section 4.4.1.)

Given a dual-feasible interior iterate (x, y, s) , a scalar $\delta > 0$, and an index set Q selected according to Rule 4.1, the basic search direction is defined by

$$\begin{pmatrix} -X_Q^{-1}S_Q & A_Q^T \\ A_Q & \delta I \end{pmatrix} \begin{pmatrix} \Delta x_Q \\ \Delta y \end{pmatrix} = \begin{pmatrix} c_Q - A_Q^T y \\ b - A_Q x_Q \end{pmatrix}, \quad (14)$$

and

$$\Delta s = -A^T \Delta y. \quad (15)$$

The normal equations are

$$(A_Q X_Q S_Q^{-1} A_Q^T + \delta I) \Delta y = b, \quad (16)$$

$$\Delta x_Q = -x_Q + X_Q S_Q^{-1} A_Q^T \Delta y. \quad (17)$$

When $(x_Q, s_Q) > 0$ and $\delta > 0$, these equations have a unique solution. We also define

$$\tilde{x}_i := \begin{cases} x_i + \Delta x_i & i \in Q, \\ 0 & i \in \mathbf{n} \setminus Q. \end{cases} \quad (18)$$

As in [WNTO12], the dual variables (y, s) are updated to (y^+, s^+) , via

$$y^+ := y + t_d \Delta y, \quad s^+ := s + t_d \Delta s = (c - A^T y^+), \quad (19)$$

with

$$t_d := \max\{\beta \bar{t}_d, \bar{t}_d - \|\Delta y\|\}, \quad (20)$$

$$\bar{t}_d := \min \left\{ 1, \min \left\{ \frac{s_i}{-\Delta s_i} \mid \Delta s_i < 0 \right\} \right\}, \quad (21)$$

where $\beta \in (0, 1)$ is an algorithm parameter. The primal update x^+ , adapted from [WNTO12],² is given by

$$x_i^+ := \begin{cases} \max\{\min\{\phi, \xi\}, x_i + t_p \Delta x_i\} & i \in Q, \\ \min \left\{ \frac{\mu_Q^+}{s_i^+}, \bar{\chi} \right\} & i \in \mathbf{n} \setminus Q, \end{cases} \quad (22)$$

with $\xi > 0$ and $\bar{\chi} > 0$ (respectively, small and large) algorithm parameters,

$$t_p := \max\{\beta \bar{t}_p, \bar{t}_p - \|\Delta y\|\}, \quad (23)$$

$$\bar{t}_p := \min \left\{ 1, \min \left\{ \frac{x_i}{-\Delta x_i} \mid \Delta x_i < 0, i \in Q \right\} \right\}, \quad (24)$$

²The “min” in the top portion of (22) places a positive lower bound on $x_i^+, i \in Q$, preventing it from being too close to zero away from a solution point; a small value of $\xi > 0$ ensures that this lower bound is not too large. The bottom portion of (22) helps centrality while keeping $x^+, i \notin Q$, bounded. The second argument of the “max” in (23) is needed to allow local quadratic convergence, which is also the motivation for the right-hand side of (26).

$$\mu_Q^+ := \frac{(x_Q^+)^T s_Q^+}{|Q|}, \quad (25)$$

and

$$\phi := \{\|\tilde{x}\}_-^2 + \|\Delta y\|^2. \quad (26)$$

Finally, we update the regularization parameter as

$$\delta^+ := \min\{\phi, \bar{\delta}\}, \quad (27)$$

where $\bar{\delta}$ is a small algorithm parameter. The above sequence of steps is well defined and can be repeated indefinitely (see details in Proposition 3.1 of [WNT012]). This justifies attaching an iteration superscript k to the variables.

Formula (27) is motivated by the fact that a small value of ϕ indicates proximity to a primal-dual solution. It enforces two key properties on the sequence $\{\delta^k\}$. First, δ^k is bounded (by $\bar{\delta}$) and remains bounded away from zero if Δy^k is, which is crucial to global convergence in case of rank-degeneracy. Second, because of the relationship (22) between x^+ and ϕ , when ϕ^k tends to zero (which is the case in the context of our local analysis), formula (27) enforces eventual upper-boundedness of δ^k by every Q -component of x^k . Hence, in particular, δ^k is upper bounded by every Q -component i of $x^k - x^*$ such that $x_i^* = 0$; the condition $|Q| > m$ imposed on $|Q|$ by Rule 4.1 implies that such i exists under the local-regularity Assumption 2 used in our local convergence analysis. We conclude that $\delta^k \leq \|x^k - x^*\|$. The latter turns out to be sufficient for our algorithm to inherit the quadratic rate of convergence enjoyed by that of [WNT012].

Algorithm 1: Regularized rPDAS

Input: *Problem data:* A, b, c ; *Initial iterate:* $y^0, s^0 := c - A^T y^0 > 0, x^0 > 0$;

Parameters: $\beta \in (0, 1), \bar{\chi} > 0, \xi \in (0, \infty], \epsilon \in (0, \infty], \bar{\delta} > 0$, and an upper bound $M \in \mathbf{n}$ on the number of constraints active at any dual-feasible point;

Step 0: *Initialization.* Set $k := 0, \delta^0 := \bar{\delta}$, choose Q^0 according to Rule 4.1;

Step 1: *Search direction computation.* Solve (14) for $(\Delta x_{Q^k}^k, \Delta y^k)$;

Step 2: *Updates.* Update dual variables using (19)-(21); compute \tilde{x}^k and ϕ^k from (18) and (26); update the Q^k component of primal variables using (22)-(24); compute $\mu_{Q^k}^+$ by (25), and update $x_{\mathbf{n} \setminus Q^k}^{k+1}$ using (22)-(24); set $\delta^{k+1} := \min\{\bar{\delta}, \phi^k\}$ and choose Q^{k+1} according to Rule 4.1; set $k := k + 1$; go back to Step 1.

This algorithm is identical to the rMPC* algorithm of [WNT012] with the ψ parameter of [WNT012] set to zero (no centering-corrector step, i.e., affine scaling) and the $\nu \geq 2$ parameter of [WNT012] set to 2 (for simplicity), *except* for the presence of the regularizing term δI in the left-hand side of (14).

4.2.2 Global convergence

Most of the analysis of [WNT012] applies here, with some simplifications due to the ψ parameter in [WNT012] being set to zero, and others due to our standing assumption that the primal problem is feasible, an assumption not made in [WNT012]. Under that assumption, feasibility of the sequence $\{y^k\}$ rules out the possibility that $b^T y^k \rightarrow \infty$, simplifying the statement of Lemma 3.3 of [WNT012]. On the other hand, in [WNT012] it is assumed that A has full row rank. Still, because the system of linear equations to be solved at each

iteration of the algorithm studied in [WNT012] involves A_Q rather than A , the sole role of that assumption is to ensure that Q can always be selected in such a way that A_Q has full row rank, as prescribed by the Q -selection rule used there. Because Rule 4.1 does not include such requirement, a full-rank assumption on A is not needed here.

In its simplified form, Lemma 3.3 of [WNT012] reads as follows.

Proposition 4.2. *Suppose Assumption 1 holds. Then $y^k \rightarrow y^* \in F^s$.*

Proof. In order to avoid duplicating material, we describe the necessary variations with the analysis of [WNT012]. There are four such variations. The first one has to do with convergence of $\{y^k\}$ and boundedness of $\{\tilde{x}^k\}$ ($\{\tilde{x}^{a,k}\}$ in [WNT012]), proved in Lemmas 3.3 and 3.4 of [WNT012], respectively. To see that the results still hold here, first note that, from (16), (17) and (18), we have

$$\hat{A}_{Q^k} \hat{D}_{Q^k}^k \hat{A}_{Q^k}^T \Delta y^k = b, \quad (28)$$

and

$$\tilde{x}_{Q^k} = -D_{Q^k}^k A_{Q^k}^T (\hat{A}_{Q^k} \hat{D}_{Q^k}^k \hat{A}_{Q^k}^T)^{-1} b, \quad (29)$$

where we have defined $\hat{A}_{Q^k} := [A_{Q^k} \ I]$, $D_{Q^k}^k := X_{Q^k}^k (S_{Q^k}^k)^{-1}$, and $\hat{D}_{Q^k}^k := \text{diag}(D_{Q^k}^k, \delta^k I)$. Now the proof of Lemma 3.3 in [WNT012] applies if its reference to equation (64a) of [WNT012] (which is our (16) with $\delta = 0$) is replaced with a reference to (28). Indeed, while A_{Q^k} may be rank-deficient, \hat{A}_{Q^k} has full row rank for all $Q \subseteq \mathbf{n}$, so Lemma 3.2 of [WNT012] can be invoked in connection with (28). Next, in view of (29), to prove boundedness of $\{\tilde{x}_{Q^k}^k\}$ it is enough to show that the matrix $D_{Q^k}^k A_{Q^k}^T (\hat{A}_{Q^k} \hat{D}_{Q^k}^k \hat{A}_{Q^k}^T)^{-1}$ is bounded by a constant independent of $\delta^k > 0$, of Q^k , and of diagonal $D_{Q^k}^k > 0$. To this end, note that

$$\begin{pmatrix} D_{Q^k}^k A_{Q^k}^T (\hat{A}_{Q^k} \hat{D}_{Q^k}^k \hat{A}_{Q^k}^T)^{-1} \\ \delta^k (\hat{A}_{Q^k} \hat{D}_{Q^k}^k \hat{A}_{Q^k}^T)^{-1} \end{pmatrix} = \hat{D}_{Q^k}^k \hat{A}_{Q^k}^T (\hat{A}_{Q^k} \hat{D}_{Q^k}^k \hat{A}_{Q^k}^T)^{-1},$$

so that

$$\|D_{Q^k}^k A_{Q^k}^T (\hat{A}_{Q^k} \hat{D}_{Q^k}^k \hat{A}_{Q^k}^T)^{-1}\| \leq \|\hat{D}_{Q^k}^k \hat{A}_{Q^k}^T (\hat{A}_{Q^k} \hat{D}_{Q^k}^k \hat{A}_{Q^k}^T)^{-1}\|.$$

Now, matrices of the form $DA^T(ADA^T)^{-1}$, with A full rank and $D > 0$ diagonal, are known to have bounded norm independent of D . (This result was proved algebraically by I.I. Dikin [Dik74]. G.W. Stewart gave an independent, geometric proof in [Ste89], where he refers to such matrices as ‘‘scaled pseudo-inverses’’.) Calling upon this result we conclude that

$$\|\tilde{x}_{Q^k}^k\| \leq R \|b\|, \quad (30)$$

with R independent of Q^k , $D_{Q^k}^k$ and δ^k (note there are only finitely many choices of Q^k). This establishes boundedness for the $\{\tilde{x}^k\}$ sequence since $\tilde{x}_{\mathbf{n} \setminus Q}^k = 0$ by (18). Boundedness of $\{x^k\}$ follows as in [WNT012].

The second variation with the analysis in [WNT012] is minor. It has to do with the first assertion in Lemma 3.5 of [WNT012], to the effect that (in the present paper’s notation) $A\tilde{x}^k = b$ for all k . Such equality does not hold here, due to the presence of $\delta^k \neq 0$ in the second block equation of (11). However, it is still the case that, given any infinite index set K such that $\Delta y^k \xrightarrow{K} 0$, it holds that $A\tilde{x}^k \xrightarrow{K} b$ (since, in view of the just-established boundedness of $\{\tilde{x}^k\}$ and of the definition (26)–(27) of δ , $\{\delta^k\}$ is bounded), which is sufficient for the use made of this result in [WNT012].

The third and fourth variations with the analysis in [WNT012] both occur in the proof of Lemma 3.7 of [WNT012]. First, in the present context, boundedness away from zero on K' of $\|\Delta y^k\| - \|\Delta y^{a,k}\|$ in the notation of [WNT012]—does not imply that of $\|\Delta s_Q^k\|$ ($= \|A_Q^T \Delta y^k\|$) because A_Q^T need not be full column rank.³ However it still implies boundedness away from zero on K' of $b^T \Delta y^k$ (see the displayed equation following (76) in [WNT012]). Indeed, boundedness away from zero on K' of $\{\delta^k\}$, say $\delta^k \geq \varepsilon'$ for all $k \in K'$, with $\varepsilon' > 0$, follows in view of definition (26)–(27) of δ , so that, for every $k \in K'$,

$$M(Q, \delta^k) := A_Q (S_Q^k)^{-1} X_Q^k A_Q^T + \delta^k I \succeq \varepsilon' I \succ 0,$$

and therefore

$$b^T \Delta y^k = (\Delta y^k)^T M(Q, \delta^k) \Delta y^k \geq \varepsilon' \|\Delta y^k\|^2 \geq \varepsilon' \beta^2 \quad \forall k \in K'.$$

This allows the proof of Lemma 3.7 of [WNT012] to proceed. The final variation, still with the proof of Lemma 3.7 of [WNT012], occurs in the proof of boundedness of $\{\Delta s^k\}$ (again, $\{\Delta s^{a,k}\}$ in [WNT012]) on K'' , where (in the last three displayed equations in that proof) the standing assumption in [WNT012] of A_Q being full row rank is invoked. In the present context, to see that Δs^k is bounded, we can just use

$$\Delta s^k = -A^T \Delta y^k = -A^T M(Q, \delta^k)^{-1} b,$$

and note that the right-hand side is bounded, since $0 \prec M(Q, \delta^k)^{-1} \preceq (\varepsilon')^{-1} I$. \square

As was the case for [WNT012, Theorem 3.8], by invoking a linear independence assumption, convergence to the optimal set can be proven. Specifically, we can use the argument of [TAW06, Lemma 11], followed by a similar argument used in [WNT012, Theorem 3.8] (see also [TAW06, Theorem 12]). Again accounting for the fact that $\{y^k\}$ is bounded, we obtain the following statement.

Theorem 4.3. *Suppose Assumption 1 holds. Under the further assumption that, at every dual-feasible point, the gradients of all active constraints are linearly independent, it holds that $y^k \rightarrow y^* \in F^*$.*

Whether or not the additional assumption is truly necessary to guarantee convergence to F^* is still an open question; this uncertainty is part of the reason why, following [WNT012], we have held out invoking this assumption until this last step. Furthermore, a similar statement to that made in [WNT012, Remark 3.1], regarding the practical expectation of convergence of the dual sequence to the optimal set also applies here.

Finally, as in [WNT012, Proposition 3.9], the following convergence properties of the primal sequence hold without further assumptions.

Proposition 4.4. *Suppose Assumption 1 holds and $y^k \rightarrow y^* \in F^*$. Then there exists an infinite index set K on which $\{\Delta y^k\}$ converges to zero and $\{\tilde{x}^k\}$ converges to the primal optimal set.*

4.2.3 Quadratic local convergence

A similar local convergence analysis to that in [TAW06] (and [WNT012]) applies here. The following assumption is used.⁴

³The proof in [WNT012] contains an unfortunate typographical error: In three places, in the text that precedes equation (76), $\Delta s^{a,k}$ should be $\Delta s_Q^{a,k}$.

⁴Note that Assumption 2 does not rule out rank degeneracy away from the solution, nor even close to the solution unless $|Q^k|$ is exactly equal to m .

Assumption 2. (i) The dual solution set is a singleton, i.e., $F^* = \{y^*\}$, (ii) $y^k \rightarrow y^*$, and (iii) the set $\{a_i \mid c_i = a_i^\top y^*\}$ is linearly independent.

Some obvious consequences of this assumption are that $s^k \rightarrow s^* := c - A^\top y^*$, that the optimal multiplier x^* associated to y^* is unique, that strict complementarity holds,⁵ i.e., that $x_i^* > 0$ for all $i \in I(y^*)$, and further that $\text{span}\{a_i \mid i \in I(y^*)\} = \mathbb{R}^m$. (This last condition shows that Assumption 2 implies that A is full row rank, an assumption we did not use in the global analysis.) Since $y^k \rightarrow y^*$, Rule 4.1 implies that $I(y^*) \subseteq Q^k$ for all k large enough (see Lemma 4.1) and so Assumption 2 also implies that A_{Q^k} has full row rank for all k large enough.

The next lemma gives further implications of Assumption 2, including the fact that, as intended, $\delta^k \rightarrow 0$.

Lemma 4.5. *Suppose Assumption 2 holds. Then $\Delta y^k \rightarrow 0$, $\tilde{x}^k \rightarrow x^*$, $x^k \rightarrow x^*$, $\Delta x_Q^k \rightarrow 0$, $\phi^k \rightarrow 0$, $\delta^k \rightarrow 0$.*

Proof. The proof of the first four assertions uses arguments very similar to those used in Lemmas A.2, A.3 and A.5, and Proposition A.4, in [WNT012, Electronic Supplemental Material (ESM)], and is omitted: the only difference occurs where the Jacobian and reduced Jacobian $J(A_Q, x_Q^k, s_Q^k)$ and $J_a(A_Q, x_Q^k, s_Q^k)$ are invoked in the proofs of Proposition A.4 and Lemma A.5. Since $\{\delta^k\}$ is bounded (see (27)), it suffices to extract a sub-sequence where it converges (to some $\delta^* \geq 0$) and to note that, under Assumption 2, the regularized versions of $J(A_Q, x_Q^k, s_Q^k)$ and $J_a(A_Q, x_Q^k, s_Q^k)$ converge to a non-singular matrix regardless of whether δ^* is positive or zero. The last two assertions are direct consequences of the convergence of Δy^k to zero and of \tilde{x}^k to $x^* \geq 0$. \square

Once we know that $\delta^k \rightarrow 0$, in view of the local convergence results established in [TAW06] and [WNT012, ESM] for similar constraint-reduced algorithms, we may expect that quadratic convergence should take place for our regularized algorithm, as long as δ^k goes to zero “fast enough”. This turns out to be the case, and local convergence is indeed quadratic, as is verified next.

Theorem 4.6. *Suppose Assumptions 2 holds. Then $\{(x^k, y^k)\}$ converges to (x^*, y^*) q -quadratically. Further, $\{t_p^k\}$ and $\{t_d^k\}$ both converge to 1.*

Proof. Following [WNT012], we use the notation $z := (x, y)$, $z^* := (x^*, y^*)$, and so on. The key difference between the present context and that of [WNT012] is the entry δI in the bottom-right block of (12), while that entry is zero in [WNT012]. Along the lines of the proof of Lemma 16 of [TAW06], which is invoked in the proof of Lemma A.8 in [WNT012, ESM], it is readily shown that

$$\begin{pmatrix} S_Q & -X_Q A_Q^\top \\ A_Q & \delta I \end{pmatrix} (\Delta z_Q - \Delta z_Q^N) = \begin{pmatrix} 0 & 0 \\ 0 & -(A_{n \setminus Q} X_{n \setminus Q} S_{n \setminus Q}^{-1} A_{n \setminus Q}^\top + \delta I) \end{pmatrix} \Delta z_Q^N. \quad (31)$$

(Compare with the equation that spans the very bottom of p.132 and very top of p.133 of [TAW06]; the only difference is the presence here of the term δI .) Assumption 2 implies that, when z is close enough to z^* , Q is such that A_Q has full row rank, making the matrix in the left-hand side of (31) non-singular, with bounded inverse. To conclude the proof of an analog of Lemma A.8 of [WNT012, ESM] (with identical statement),

⁵This follows from Assumption 2 because solvable linear programs always have at least one strictly complementary solution [Wri97, p.28].

it suffices to show that the new term in the right-hand side of (31), namely δI , is of the order of $\|z - z^*\|$ when z is close enough to z^* . To this effect, note that, from (27),

$$\delta \leq \phi^-, \quad (32)$$

where ϕ^- indicates the value of ϕ from the previous iteration. On the other hand, since ξ is strictly positive and since $\phi^k \rightarrow 0$, (22) implies that $x_i \geq \phi^-$ for all $i \in Q$, and hence, when z is close enough to z^* ,

$$\phi^- \leq \min_{i \in Q} x_i \leq \min_{\{i \in Q \mid x_i^* = 0\}} |x_i - x_i^*| \leq \|x - x^*\| \leq \|z - z^*\|. \quad (33)$$

Indeed, $\{i \in Q \mid x_i^* = 0\}$ is nonempty since Rule 4.1 together with Assumption 2 ensures that Q contains at least one constraint that is inactive at the solution. Finally, (32) and (33) then imply that for all z close enough to z^* ,

$$\delta \leq \|z - z^*\|. \quad (34)$$

Hence Lemma A.8 of [WNT012, ESM] holds in the present context. Most of the remainder of the proof of quadratic convergence follows [WNT012, ESM], with nontrivial changes being needed only in the last two pages, in “Proof of Theorem 4.1”, to account for the fact that, in contrast with the situation in [WNT012], the “constraint-unreduced” primal-dual search direction $\Delta z(z, \mathbf{n})$ is not identical to the Newton step $d^N(z)$ for the solution of the system

$$\begin{pmatrix} X(c - A^T y) \\ Ax - b \end{pmatrix} = 0. \quad (35)$$

Specifically, from (12) and with $s := c - A^T y$,

$$H(z, \delta) \Delta z(z, \mathbf{n}) = \begin{pmatrix} Xs \\ b - Ax \end{pmatrix}, \quad (36)$$

while

$$H(z, 0) d^N(z) = \begin{pmatrix} Xs \\ b - Ax \end{pmatrix}, \quad (37)$$

where

$$H(z, \delta) := \begin{pmatrix} -S & XA^T \\ A & \delta I \end{pmatrix}. \quad (38)$$

Note that $H(z, \delta)$ is invertible for z close to z^* , and $\delta \geq 0$.

In [WNT012, ESM], the identity of $\Delta z(z, \mathbf{n})$ and $d^N(z)$ is used at three places: (i) in line 5 of p.8, “it will follow” is subject to validity of the inequality

$$\|\Delta z(z, \mathbf{n})\| \leq c \|d^N(z)\| \quad (39)$$

for some $c > 0$, for all z close to z^* ; (ii) in the penultimate displayed equation on p.8, last inequality, the

bounding of the second term will still be valid if

$$\|z + \Delta z(z, \mathbf{n}) - z^*\| \leq d\|z - z^*\|^2 \quad (40)$$

for some $d > 0$ for all z close to z^* , with $(x, s) \geq 0$; and (iii) in the last displayed equation on p.9, last inequality, the bounding of the second term will still be valid if

$$\|z_Q + \Delta z_Q(z, \mathbf{n}) - z_Q^*\| \leq d\|z - z^*\|^2, \quad (41)$$

again for some $d > 0$ and all z close to z^* . Our proof is then concluded by noting that (41) is a direct consequence of (40), that (39) follows readily from (36) and (37), and that (40) follows again from (36) and (37) and from the bound (34) on δ . Specifically, concerning the latter,

$$\|z + \Delta z(z, \mathbf{n}) - z^*\| \leq \|z + d^N(z) - z^*\| + \|\Delta z(z, \mathbf{n}) - d^N(z)\|,$$

and, in view of (34) and of local Lipschitz continuity of $H(z, \delta)^{-1}$ for z close to z^* , and $\delta \geq 0$, the requisite bound on the second term follows, using (36) and (37) to express

$$\Delta z(z, \mathbf{n}) - d^N(z) = (H(z, \delta)^{-1} - H(z, 0)^{-1}) \begin{pmatrix} Xs \\ b - Ax \end{pmatrix}, \quad (42)$$

and from the fact that, since $X^*s^* = 0$ and $Ax^* = b$,

$$\left\| \begin{pmatrix} Xs \\ b - Ax \end{pmatrix} \right\| = \left\| \begin{pmatrix} X(s - s^*) + S^*(x - x^*) \\ -A(x - x^*) \end{pmatrix} \right\| \leq d\|z - z^*\| \quad (43)$$

for some $d > 0$, for z close to z^* . □

4.2.4 Regularization in the limit of small δ : motivating the Kernel-Step algorithm

Consider the Δy component of the search direction of Algorithm 1, which satisfies the regularized normal equation (16). For simplicity of notation, we take $Q = \mathbf{n}$. We investigate what happens in the limit of small regularization parameter δ . Consider a spectral decomposition of the unregularized normal matrix

$$M := AS^{-1}XA^T = V\Sigma V^T = \begin{pmatrix} V_1 & V_2 \end{pmatrix} \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}$$

with the columns of V_1 spanning $\mathcal{R}(M)$, the columns of V_2 spanning $\mathcal{N}(M)$, and $\Sigma_1 \succ 0$. The dual step then can be expressed as

$$\Delta y = (V\Sigma V^T + \delta VV^T)^{-1}b = V_1(\Sigma_1 + \delta I)^{-1}V_1^T b + \delta^{-1}V_2V_2^T b. \quad (44)$$

As $\delta \rightarrow 0$, the first term of (44) converges to the least norm solution of $M\Delta y = b$, and the second term is the projection of b onto $\mathcal{N}(A^T) = \mathcal{N}(M)$ scaled by δ^{-1} . So, for vanishing δ , the direction of Δy will be along the projection of b onto the nullspace of A^T , unless b is orthogonal to this nullspace, in which case the

limit of Δy will be the least norm solution to the normal equations. Next we propose an algorithm based on this limit search direction.

4.3 Kernel-Step rPDAS

4.3.1 Algorithm details

This algorithm is similar to Algorithm 1, and we are able to prove similar convergence results. Again, the analysis borrows from arguments used in [WNT012], which itself borrows from [TAW06]. As for the Regularized algorithm, we start from a dual strictly feasible point y^0 , and primal interior point $x^0 > 0$, not necessarily satisfying $Ax^0 = b$. We again select Q according to Rule 4.1. Next, we determine whether $b \in \mathcal{R}(A_Q)$. Specifically, we compute the orthogonal projector P onto the kernel of A_Q^T via QR factorization of A_Q , and declare b to be in $\mathcal{R}(A_Q)$ if $\|Pb\| \leq \eta$, $\eta > 0$ a small prescribed value. If so, we take a “regular” step defined to be along the least norm solution to

$$(A_Q S_Q^{-1} X_Q A_Q^T) \Delta y = b, \quad (45)$$

i.e.,

$$\Delta y = (A_Q S_Q^{-1} X_Q A_Q^T)^\dagger b. \quad (46)$$

where \dagger denotes the Moore-Penrose pseudo-inverse. If, instead, $b \notin \mathcal{R}(A_Q)$, then we take a “kernel” step along a direction Δy defined to be the orthogonal projection of b onto $\mathcal{N}(A_Q^T)$, so that $\Delta s_Q = 0$. In either case, the primal update is defined in the same way as in Algorithm 1 using equations (17) and (22)-(24), so that, by (18), $\tilde{x} = 0$ for kernel steps. The dual updates are again defined by (19), with t_d selected as follows. In the case of a regular step, the dual step length t_d is chosen according to (20)-(21). In the case of a kernel step, we define t_d differently. First of all, the kernel step length is not limited to 1; we go along this direction until we encounter a blocking constraint not in working set Q , and take a step almost all of the way to the boundary. (This does not hinder the local quadratic rate of convergence obtained in the local convergence analysis of section 4.3.3, because as shown below (Lemma 4.12) at most finitely many kernel steps will be taken.) Specifically, defining

$$\hat{i} := \arg \min \left\{ \frac{s_i}{-\Delta s_i} \mid \Delta s_i < 0 \right\}, \quad (47)$$

(Proposition 4.9 below shows that \hat{i} is always well defined) and

$$\bar{s} := \min\{\bar{s}_M, \epsilon\}, \quad (48)$$

where $\epsilon > 0$ and M are the defining parameters for $\mathcal{Q}_{\epsilon, M}(y)$, and \bar{s}_M is an M th smallest entry of s , we chose t_d as

$$t_d := \frac{s_{\hat{i}} - \theta \bar{s}}{|\Delta s_{\hat{i}}|}, \quad (49)$$

where $\theta \in (0, 1)$ is an algorithm parameter. This choice of t_d yields

$$s_{\hat{i}}^+ = \theta \bar{s} \quad (50)$$

for the updated value of $s_{\hat{i}}$, ensuring that the blocking constraint (indexed by \hat{i}) will be contained in the updated working set Q^+ , and thereby preventing more than a finite number of kernel steps from being taken (Lemma 4.12). Also, for future reference, note that, since $\Delta_{s_Q} = 0$, $\hat{i} \notin Q$.

We now state the Kernel-Step rPDAS algorithm.

Algorithm 2: Kernel-Step rPDAS

Input: *Problem data:* A, b, c ; *Initial iterate:* $y^0, s^0 := c - A^T y^0 > 0, x^0 > 0$;

Parameters: $\beta \in (0, 1)$, $\bar{\chi} > 0$, $\xi \in (0, \infty]$, $\epsilon \in (0, \infty]$, $\theta \in (0, 1)$, $\eta > 0$, and an upper bound $M \in \{m + 1, \dots, n\}$ on the number of constraints active at any dual-feasible point;

Step 0: *Initialization.* Set $k := 0$, choose Q^0 according to Rule 4.1;

Step 1a: *Regular step. Direction computation and dual update.* If $b \in \mathcal{R}(A_{Q^k})$ (regular step), compute Δy^k as the least norm solution of (16) with $\delta := 0$ and update the dual variables using (19)-(21) and go to Step 2.. Else go to Step 1b.

Step 1b: *Kernel step. Direction computation and dual update.* Set Δy^k equal to the orthogonal projection of b onto $\mathcal{N}(A_{Q^k}^T)$ and update the dual variables using (19), (47)-(49).

Step 2: *Updates.* Compute $\Delta x_{Q^k}^k, \tilde{x}^k$, and ϕ^k from (17), (18), and (26), update the Q^k component of the primal variables using (22)-(24), compute $\mu_{Q^k}^+$ from (25), and compute $x_{n \setminus Q^k}^+$ using (22)-(24), choose Q^{k+1} according to Rule 4.1, set $k := k + 1$ and go to Step 1.

Remark 1. The Kernel-Step algorithm admittedly involves operations that are known to be numerically delicate, namely, checking whether a vector b is in the range of a matrix A_Q , and if not, projecting b onto the kernel of A_Q^T . This does not lessen the practical value of the algorithm however, as its behavior is not highly sensitive to inaccuracies in these calculations. Indeed, the purpose of the kernel step is to generate an iterate away from the current one, along a direction in the (polyhedral) “tube” formed by the columns of A_Q , intersecting a constraint boundary associated to a column of A not in A_Q before one associated to a column of A_Q . While this will *always* be the case if the kernel direction is computed exactly (so that the direction will be parallel to the faces of the tube), it will often still be so with inexact computation, even when the error is significant.

Remark 2. In its proposed implementation, every iteration of the Kernel-Step algorithm involves a QR factorization of A_Q , whose numerical cost tends to prohibitively dominate when $|Q| \gg m$ with A sparse. (See Figure 6 below.) (In practice though, a size $|Q|$ of the order of $2m$ or $3m$ seems to work well.) This high cost when $|Q| \gg m$ could possibly be mitigated by proceeding as follows: when it is deemed unlikely that the “normal” matrix will be rank-deficient, first form that matrix and attempt its Cholesky factorization; only resort to carrying out a QR factorization of A_Q when the Cholesky factorization fails.

4.3.2 Global convergence

First we note that if Δy comes from a kernel step, then it cannot be arbitrarily small.

Lemma 4.7. *There exists $\gamma > 0$, depending only on A and b , such that $\|\Delta y\| \geq \gamma$ whenever Δy corresponds to a kernel step.*

Proof. A kernel step is taken only when b is not in the range of A_Q , thus not orthogonal to the kernel of A_Q^T . Since kernel steps Δy are obtained by projection of b on the kernel of A_Q^T , they are nonzero. Since there

are only finitely many different such Δy (because there are only finitely many sub-matrices A_Q), there must exist $\gamma > 0$ such that $\|\Delta y\| \geq \gamma$. \square

As in the analysis of the Regularized rPDAS, monotonicity of $\{b^\top y^k\}$ plays a key role. The first lemma shows that Δy generated by the Kernel-Step algorithm is always an ascent direction.

Lemma 4.8. *Let Δy be as constructed by Algorithm 2. Then $\Delta y \neq 0$ and $b^\top \Delta y > 0$.*

Proof. If $b \in \mathcal{R}(A_Q)$, under our assumption that $b \neq 0$, $\Delta y = 0$ is not a solution to equation (45), so the least norm solution Δy is nonzero; and if $b \notin \mathcal{R}(A_Q)$, then b is not orthogonal to $\mathcal{N}(A_Q^\top)$ and hence its orthogonal projection Δy on $\mathcal{N}(A_Q^\top)$ again is nonzero. As for the second claim, if $b \notin \mathcal{R}(A_Q)$, it is immediate, and if $b \in \mathcal{R}(A_Q)$, positive semi-definiteness of $A_Q S_Q^{-1} X_Q A_Q^\top$ implies that

$$(\Delta y)^\top b = (\Delta y)^\top (A_Q S_Q^{-1} X_Q A_Q^\top) \Delta y = \|(A_Q S_Q^{-1} X_Q A_Q^\top)^{1/2} \Delta y\|^2 > 0.$$

Indeed, $(A_Q S_Q^{-1} X_Q A_Q^\top)^{1/2} \Delta y$ cannot vanish, since $(A_Q S_Q^{-1} X_Q A_Q^\top) \Delta y = b \neq 0$. \square

Under our standing assumptions, Algorithm 2 is well defined.

Proposition 4.9. *Suppose Assumption 1 holds. Then Algorithm 2 is well defined, and constructs quantities with the properties that $t_d \in (0, \infty)$, $y^+ \in F^\circ$, $s^+ = c - A^\top y^+ > 0$, and $x^+ > 0$.*

Proof. First, since $s > 0$, it always holds that $\bar{t}_d > 0$. Since $b^\top \Delta y > 0$ (Lemma 4.8), it follows from non-emptiness of the solution set, which is implied by Assumption 1, that the largest feasible dual step size must be finite. This implies, in view of (19), that $\Delta s_i < 0$ for some i , so t_d is always finite and \hat{i} is well defined. It is also easy to see that, under a kernel step, $\Delta s_Q = 0$, so we must have $\hat{i} \in \mathbf{n} \setminus Q$. In view of (19), it is always the case that $s^+ = c - A^\top y^+$. It remains to show that $t_d > 0$ and that $s^+ > 0$ (i.e., $y^+ \in F^\circ$) and $x^+ > 0$. When $b \in \mathcal{R}(A_Q)$, $t_d > 0$ follows from (21) and the fact that $\bar{t}_d > 0$. When $b \notin \mathcal{R}(A_Q)$, $t_d > 0$ follows from (49) and the facts that $|\Delta s_i| > 0$, $s_i > 0$, $\theta \in (0, 1)$, $\bar{s} \leq s_i$ (since $\hat{i} \in \mathbf{n} \setminus Q$). Next, the inequality $s^+ > 0$ follows from (20) and (19) when $b \in \mathcal{R}(A_Q)$, while, when $b \notin \mathcal{R}(A_Q)$, we have for all i ,

$$s_i^+ = s_i + t_d \Delta s_i = s_i \left(1 + t_d \frac{\Delta s_i}{s_i} \right) \geq s_i \left(1 + t_d \frac{\Delta s_{\hat{i}}}{s_i} \right) = s_i \left(1 - \frac{s_i - \theta \bar{s}}{s_i} \right) > 0,$$

where we have used definitions (47) of \hat{i} and (49) of t_d , and the facts that $\Delta s_i < 0$ (from (47)) and $\theta \bar{s} > 0$. Finally, because $\Delta y \neq 0$ (Lemma 4.8), (22) and (26) imply that for $i \in Q$, $x_i^+ > 0$, while for $i \in \mathbf{n} \setminus Q$, (22), (25), $x_Q^+ > 0$, and $s^+ > 0$, again give $x_i^+ > 0$. \square

Our analysis again focuses on the dual sequence $\{y^k\}$. In view of Lemma 4.8 and Proposition 4.9, $\{b^\top y^k\}$ is strictly increasing and $y^k \in F^\circ$. It follows from primal feasibility (Assumption 1) that $\{b^\top y^k\}$ is also bounded. This again implies that the dual sequence converges and the primal sequence is bounded. To prove the former, we make use of the following simple extension of a result in [Sai96]; see also [HT12].

Lemma 4.10. *Given a matrix A and a vector b in the range of A , there exists a constant $C > 0$ such that, for any positive definite diagonal matrix D , the least-squares solution to*

$$ADA^\top \Delta y = b$$

satisfies $\|\Delta y\| \leq Cb^\top \Delta y$.

Proof. Let $A = QR$ be a QR-decomposition of A . We have

$$QRDR^\top Q\Delta y = b. \quad (51)$$

Write $R^\top = [R_1^\top \ 0]$, where R_1 has full row rank. Also let $v := Q\Delta y$, where Δy is the least-squares solution of (51), and write $v^\top = [v_1^\top \ v_2^\top]$ where the dimension of v_1 is equal to the rank of R_1 . We have $R^\top Q\Delta y = R_1^\top v_1$ and $RDR_1^\top v_1 = Q^\top b$. Since v_2 is free, the least-squares property of Δy implies that $v_2 = 0$. Further, compatibility of the system (b in the range of A) implies that $R_1DR_1^\top v_1 = Q_1^\top b$, where Q_1^\top is the upper portion of Q^\top . Theorem 5 in [Sai96] (see also Lemma 2.1(i) in [HT12]) then implies that (using the fact that $v_2 = 0$)

$$\|\Delta y\| = \|v\| = \|v_1\| \leq C|b^\top Q_1 v_1| = C|b^\top Qv| = C|b^\top \Delta y|,$$

where C only depends on R_1 , Q and b , i.e., only on A and b . The proof is complete. \square

Lemma 4.11. *Suppose Assumption 1 holds. Then $y^k \rightarrow y^*$ for some $y^* \in F$ and $\{s^k\}$ converges.*

Proof. Since $\{b^\top y^k\}$ is increasing and bounded, we have

$$\sum_{k=0}^{\infty} |b^\top (y^{k+1} - y^k)| = \sum_{k=0}^{\infty} b^\top (y^{k+1} - y^k) < \infty, \quad (52)$$

To complete the proof of the first claim, we show that there is a constant C such that

$$\|y^{k+1} - y^k\| < C|b^\top (y^{k+1} - y^k)| \quad \forall k, \quad (53)$$

which, together with (52), implies that $\{y^k\}$ is Cauchy, hence converges. Since $y^{k+1} - y^k = t_d^k \Delta y^k$ and $b^\top (y^{k+1} - y^k) = t_d^k b^\top \Delta y^k$, (53) is equivalent to

$$\|\Delta y^k\| < C|b^\top \Delta y^k|, \quad (54)$$

If a regular step is taken from y^k , (54) follows directly from Lemma 4.10, while if a kernel step is taken from y^k , i.e., $b \notin \mathcal{R}(A_{Q^k})$, (54) is again guaranteed to hold in view of the fact that (due to finiteness of the set of possible Q^k) the angle between b and its projection Δy^k is bounded away from 90 degrees. Finally, the second claim follows from (19). \square

As a direct consequence of convergence of the dual sequence, the sequence of kernel steps is finite.

Lemma 4.12. *Suppose Assumption 1 holds. Then the sequence of kernel steps is finite.*

Proof. Proceeding by contradiction, suppose that, for some infinite index set K , a kernel step is taken from every y^k , $k \in K$. Without loss of generality, \hat{i} as defined in (47) (blocking constraint) is independent of k , $k \in K$. Such \hat{i} satisfies, for all $k \in K$, (i) $\hat{i} \notin Q^k$, implying (see Rule 4.1) that

$$s_{\hat{i}}^k \geq \bar{s}_M^k \quad \text{or} \quad s_{\hat{i}}^k > \epsilon, \quad (55)$$

and (ii) in view of (50), $s_i^{k+1} = \theta \min\{\bar{s}_M^k, \epsilon\}$, yielding

$$s_i^{k+1} \leq \theta \bar{s}_M^k, \quad (56)$$

and

$$s_i^{k+1} \leq \theta \epsilon. \quad (57)$$

First suppose that $s_i^k > \epsilon$ for all $k \in K$, k large enough. Then, taking this inequality together with (57) and letting $k \rightarrow \infty$, $k \in K$, we get (denoting by s_i^* the limit of convergent sequence $\{s_i^k\}$; see Lemma 4.11) $\epsilon \leq s_i^* \leq \theta \epsilon$, which is impossible since $\epsilon > 0$, and $\theta \in (0, 1)$. Hence, in view of (55), $s_i^k \geq \bar{s}_M^k$ must hold on an infinite subsequence and, without loss of generality, for all $k \in K$. Taking this inequality together with (56) and letting $k \rightarrow \infty$, $k \in K$ then yields $\bar{s}_M^* \leq s_i^* \leq \theta \bar{s}_M^*$, where \bar{s}_M^* is the limit of convergent sequence $\{\bar{s}_M^k\}$. Since $\bar{s}_M^* \geq 0$ and $\theta \in (0, 1)$, we conclude that $\bar{s}_M^* = s_i^* = 0$. In view of Lemma 4.1, this contradicts the fact that, for all $k \in K$, $\hat{i} \notin Q^k$. This completes the proof. \square

Hence, for k large enough, Algorithm 2 reduces to the algorithm analyzed in [WNT012] (with ψ set to 0, implying $\gamma = 0$, in the latter), with just one difference: a regular step is taken here even when A_{Q^k} is rank-deficient, provided b belongs to its range. There are three instances where the full-rank assumption on A_{Q^k} is used in the global convergence analysis of [WNT012]. The first one is in the proof of boundedness of $\{\tilde{x}^k\}$ (denoted $\{\tilde{x}^{a,k}\}$ in [WNT012]) in Lemma 3.4. This result still holds in the present context.

Lemma 4.13. *Suppose Assumption 1 holds. Then $\{\tilde{x}^k\}$ is bounded.*

Proof. Since the sequence of kernel steps is finite, there is no loss of generality in assuming that all steps are regular. We first show that $\tilde{x}_{Q^k}^k$ is bounded. Defining $D_{Q^k}^k := X_{Q^k}^k (S_{Q^k}^k)^{-1}$, from the definition (46) of the regular step and from (17)-(18), we have

$$\tilde{x}_{Q^k}^k = D_{Q^k}^k A_{Q^k}^T (A_{Q^k} D_{Q^k}^k A_{Q^k}^T)^{\dagger} b. \quad (58)$$

In general, for $M \succeq 0$, when $b \in \mathcal{R}(M)$, it holds that $(M + \delta I)^{-1} b \rightarrow M^{\dagger} b$ as $\delta \rightarrow 0$ (cf. section 4.2.4). Together with the result on boundedness of “scaled pseudo-inverse” matrices, proven in [Ste89] and discussed in section 4.2.2 (see (29) and (30)), that for some R that depends only on A ,

$$\|D_{Q^k}^k A_{Q^k}^T (A_{Q^k} D_{Q^k}^k A_{Q^k}^T + \delta I)^{-1} b\| \leq R \|b\| \quad \forall \delta > 0,$$

(58) establishes boundedness of the $\{\tilde{x}^k\}$ sequence (since again $\tilde{x}_{n \setminus Q^k}^k := 0$). \square

The second and third instances where the full-rank assumption is invoked in [WNT012] occur in the proof of Lemma 3.7 in that paper. These instances are readily resolved (see below) and, like in [WNT012], convergence to the set of dual-stationary points follows.

Proposition 4.14. *Suppose Assumption 1 holds. Then $y^k \rightarrow y^* \in F^s$.*

Proof. Lemma 3.7 of [WNT012] is key. In its proof, the full rank assumption is invoked twice, within an argument where it is assumed that, for a certain infinite index set K' , $\|\Delta y^k\|$ ($\|\Delta y^{a,k}\|$ in [WNT012]) is bounded, Q^k is constant and equal to Q , and x_i^k is bounded away from zero for all $i \in Q$. The proof

extends to the present context in both cases; a key role is played by the fact that Δy^k being the least norm solution of the normal equation implies that $\Delta y^k \perp \mathcal{N}(A_Q^T)$. First, the full rank assumption is invoked when proving that $\|\Delta s_Q^k\|$ is bounded away from zero on K' . In the present context this result follows directly from $\Delta s_Q^k = A_Q^T \Delta y^k$, $\Delta y^k \perp \mathcal{N}(A_Q^T)$, since $\|\Delta y^k\|$ is bounded away from zero on K' . Second, it is invoked in proving that $\|\Delta s^k\|$ is bounded on K' . Here, since $\Delta y^k \perp \mathcal{N}(A_Q^T)$, we can solve for Δy^k in (17) (using (18)), and then compute Δs^k from (15) to get

$$\Delta s^k = -A^T(A_Q^T)^\dagger X_Q^{-1} S_Q \tilde{x}_Q^k,$$

and note that the right-hand side is bounded on K' , since \tilde{x}^k is bounded and the components of x^k are bounded away from zero on K' . This completes the proof of [WNT012, Lemma 3.7] for the present context. Convergence of $\{y^k\}$ to a point in F^s follows exactly like it does in the proof of [WNT012, Theorem 3.8] (using the known boundedness of that sequence). \square

Finally, as in the case of Algorithm 1, the following results hold.

Theorem 4.15. *Suppose Assumption 1 holds. Under the further assumption that, at every dual-feasible point, the gradients of all active constraints are linearly independent, it holds that $y^k \rightarrow F^*$.*

Proposition 4.16. *Suppose Assumption 1 holds. Then there exists an infinite index set K on which Δy^k converges to zero and $\{\tilde{x}^k\}$ converges to the primal optimal set.*

4.3.3 Quadratic local convergence

We again can prove q-quadratic convergence of $\{z^k\} := \{(x^k, y^k)\}$ generated by the Kernel-Step rPDAS algorithm under Assumption 2. As before, let y^* denote the unique solution to (2), i.e., $F^* = \{y^*\}$, let $s^* := c - A^T y^*$, and let x^* be the corresponding multiplier vector, unique in view of Assumption 2.

Theorem 4.17. *Suppose Assumption 2 holds. Then $\{(x^k, y^k)\}$ converges to (x^*, y^*) q-quadratically. Further, $\{t_p^k\}$ and $\{t_d^k\}$ both converge to 1.*

Proof. In view of Rule 4.1, Assumption 2 also ensures that A_{Q^k} has full row rank for all k large enough. It follows that, for $k \geq k_0$, for some k_0 , the sequences $\{x^k\}$ and $\{y^k\}$ are identical with those produced by setting parameter $\psi = 0$ in rMPC* of [WNT012], with x^{k_0} and y^{k_0} as initial primal and dual iterates. The local convergence analysis of [WNT012] then applies, so that the sequence $\{(x^k, y^k)\}$ converges q-quadratically. (See [WNT012, Theorem 4.1], and also [TAW06, Theorem 17].) \square

4.4 Extensions

4.4.1 Predictor-corrector variants

With some amount of work, a barrier term and a Mehrotra-type corrector component can be added to either of the algorithms discussed in this paper. This can be done in a similar way as was done for the unregularized case in [WNT012], where the analysis of [TAW06] was modified to allow such an augmentation to the algorithm, preserving the convergence result. Such addition is expected to improve practical performance significantly on many problem instances, similar to what was observed in [WNT012]. We leave the details for future work.

4.4.2 Infeasible starting point

Algorithms 1 and 2 can be extended to allow for dual-infeasible starting points, by adapting the (ℓ_1 or ℓ_∞) exact-penalty-function scheme proposed and analyzed in [HT12]. Specifically, in the ℓ_1 case, Algorithms 1 or 2 is applied instead to the problem

$$\max b^T y - \rho e^T z \quad \text{s.t.} \quad A^T y - z \leq c, \quad z \geq 0.$$

At the end of each iteration, (just before k is incremented,) penalty parameter $\rho > 0$ is updated with the rule $\rho^{k+1} := \sigma \rho^k$ (where $\sigma > 1$ is prescribed) *if and only if* either

$$z^{k+1} \geq \gamma_1 \frac{\|z^0\|}{\rho^0} \rho^k$$

or

$$\left\| \begin{pmatrix} \Delta y^k \\ \Delta z^k \end{pmatrix} \right\| \leq \frac{\gamma_2}{\rho^k} \quad \text{AND} \quad \tilde{x}_Q^k \geq -\gamma_3 e \quad \text{AND} \quad \tilde{u}_Q^k \not\leq \gamma_4 e$$

(or both) holds, where γ_1 through γ_4 are prescribed positive constant, and u is the dual variable associated to $z \geq 0$.

The analysis in [HT12] can be extended to the present situation indeed. Lemmas 3.1, 3.2, and 3.4, and Proposition 3.5 of [HT12] apply without changes. The result of Lemma 3.3 also holds unchanged, but its proof needs to be adapted. Specifically, because (3) of [HT12] must be replaced with (28), equation (21) of [HT12], invoked in the proof, must be modified by changing the last factor in the expression defining H_k as follows: replace A^Q with \hat{A}_Q , X^Q and S^Q with $\hat{X}_Q := \text{diag}(X_Q, \delta^{1/2} I_m)$ and $\hat{S}_Q := \text{diag}(S_Q, \delta^{-1/2} I_m)$, and E^Q with $\hat{E}_Q := [E_Q \quad 0_m]$, where I_m and 0_m are the identity and zero matrices of size $m \times m$. Note however that equation (38) of [HT12] remains unchanged, as do the first two factors in the expression for H_k . Equation (4) then follows, and the proof of Lemma 3.3 proceeds as in [HT12], but with reference to condition (17) of [HT12] replaced with a reference to the primal update rule in Algorithms 1 and 2. The following convergence results ensue (see Proposition 4.1 and Theorem 4.3 in [HT12]).

Proposition 4.18. *Suppose Assumption 1 holds. Then ρ is updated no more than finitely many times, and the sequence $\{(y^k, z^k)\}$ generated by the ℓ_1 or ℓ_∞ penalized version of Algorithm 1 or 2 converges to a stationary point of the penalized problem associated with the final value of ρ .*

Theorem 4.19. *Suppose Assumption 1 holds and, for all $y \in \mathbb{R}^m$, $\{a_i | a_i^T y = c_i\}$ is a linearly independent set of vectors. Then the sequence $\{(y^k, z^k)\}$ generated by the ℓ_1 or ℓ_∞ penalized version of Algorithm 1 or 2 converges to $(y^*, 0)$, with $y^* \in F^*$. Further, if Assumption 2 holds, then convergence is q -quadratic.*

5 Numerical experiments

In this section we investigate the performance of the Regularized and Kernel-Step rPDAS algorithms on some especially degenerate test problems, and observe some qualitative behavior.

5.1 Implementation

Algorithms 1 and 2 were implemented in Matlab 7.14 and run on an Intel(R) Core i7 Laptop within a virtual (VMware workstation 9) 32-bit Ubuntu 12.04 Linux guest machine with 4GB of RAM. To compute the search directions for Regularized rPDAS, we solved the regularized reduced normal equations (6) using Matlab’s Cholesky factorization routine `chol` and in case numerical problems caused this factorization to fail, we attempt a symmetric-indefinite factorization of the regularized augmented system (5) using Matlab’s `ldl` routine. Parameters for Algorithm 1, Regularized rPDAS, were chosen as $\beta := 0.95$, $\bar{\chi} := 10^9$, $\xi := 10^{-4}$, and $\bar{\delta} := 10^{-6}$ and for each problem discussed below, we assumed that a reasonably small upper bound M on the number of active constraints was available, and so we always took $\epsilon := \infty$. For the Kernel-Step rPDAS, we used the same parameters as we used for the Regularized rPDAS, and additionally set $\theta := 0.5$ and $\eta = 10^{-9}$. At each iteration we computed a basis for the kernel of A_Q^T using Matlab’s QR factorization, specifically, using

```
[Q,R,E]=qr(full(A(:,Q)),0);
i=find(abs(diag(R))<zero_thresh,1,'first');
Z=Q(:,i:end);
```

with `zero_thresh` set to 10^{-12} . This is somewhat faster than using Matlab’s `null` routine (based on a singular value decomposition) on A_Q^T , for the type and size of matrices dealt with here, and is also faster and more reliable than using a QR factorization or `null` on the normal matrix. The projection of b onto the kernel of A_Q^T is given by $ZZ^T b$.

For both algorithms, on each problem tested, the code was supplied with strictly dual-feasible initial points (i.e., $y^0 \in F^o$) as specified below, and we set $x^0 := e$, the vector of ones. We used a stopping criterion adapted from [Meh92, p. 592] (and identical to that used in [WNT012]), based on normalized constraint violation and duality gap. Specifically, convergence was declared when

$$\text{termcrit} := \max \left\{ \frac{\|b - Ax\|}{1 + \|x\|}, \frac{|c^T x - b^T y|}{1 + |b^T y|} \right\} < \text{tol}, \quad (59)$$

where `tol` was set to 10^{-8} . (Note that the first, third, and fourth terms in the max in the expression used in [WNT012] are zero here, and were left out.)

Next, we introduce a class of problem where the level of rank-degeneracy can be precisely controlled.

5.2 The tube-in-cube class of problems

The procedure for creating a problem in this class is as follows. First, generate A , $b \sim \mathcal{N}(0, 1)$ and normalize the columns of A . Next, project the columns of A onto a random $r \leq m$ dimensional subspace, so that A now has rank r . (We selected the random subspace as the range of a random $m \times r$ matrix with elements sampled from $\mathcal{N}(0, 1)$ and constructed a projector onto this subspace using Matlab’s QR factorization routine.) We refer to the resulting A matrix as “tube” matrix and denote it A_{tube} . To construct an initial dual-feasible iterate, we set $y_0 := 0$, and sample $s_{\text{tube}} \sim \mathcal{U}(0, 1)$ (insuring positive values) and set $c_{\text{tube}} := A^T y_0 + s_{\text{tube}}$. To ensure the problem has a solution, we next append the “cube” bound constraints of the form $\|y\|_\infty \leq R$ by setting $A_{\text{cube}} := (I_m, -I_m)$, with I_m the $m \times m$ identity matrix, $c_{\text{cube}} := s_{\text{cube}} := R e_{2m}$, where e_{2m} is

the $2m$ -vector of ones, and finally setting $A := (A_{\text{cube}}, A_{\text{tube}})$, $c := (c_{\text{cube}}, c_{\text{tube}})$ and $s_0 := (s_{\text{cube}}, s_{\text{tube}})$. As mentioned above, the initial primal iterate is set to $x_0 := e_n$.

This problem class was named the “tube-in-cube” in [TAO06], because of the geometry of the resulting dual-feasible region which is illustrated in Figure 2 for a 2-dimensional problem with two “tube” constraints (i.e., $m = 2$, $n = 2m + 2$).

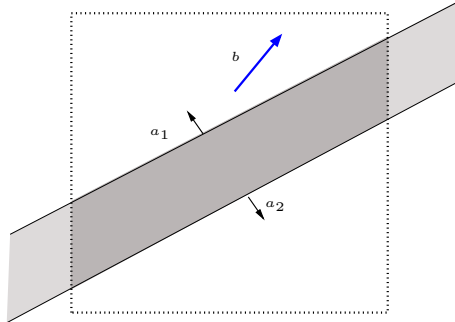


Figure 2: Tube-in-a-cube problem example in two dimensions with two “tube” constraints ($m = 2$, $n = 2m + 2$). For this problem class, the rank of the tube matrix A_{tube} is reduced to a prescribed level r which creates a $k = m - r$ dimensional “tube” in the feasible region ($m = 2$, $r = k = 1$ in the figure). Then a set of bound constraints (the “cube”) are added to ensure solvability.

We can guess how the Kernel-Step rPDAS algorithm will behave on this problem class. Looking at Figure 2, it seems likely that if the algorithm starts somewhere in the middle of the feasible region, the first step will be a kernel step that moves the iterate to the top-right part of the feasible set, and the successive iterates will be regular steps that converge toward the solution. We can imagine that if the “tube” is k -dimensional, then the algorithm will have to take k kernel steps to get to a non-degenerate “neighborhood” of the optimal set. In view of the discussion in section 4.2.4, which identifies the kernel step as a limiting case of a regularized step with vanishing regularization parameter δ , we may expect the Regularized algorithm to behave similarly for small enough $\bar{\delta}$ (from (27)).

The intuition given above turns out to be an accurate prediction of the actual behavior of the two algorithms on this problem class. Figure 3 plots the average number of “kernel-like” steps (over 10 runs) for varying degrees of rank-degeneracy (or size of the “tube-dimension”), with “cube” constraint $\|y\|_\infty \leq R$, $R = 100$. In view of (44), we deem a particular iteration of the Regularized algorithm “kernel-like” whenever,

$$\|V_2 V_2^T b\| > \alpha \|b\|, \quad (60)$$

with α a small, empirically determined constant, and the columns of V_2 form an orthonormal basis for $\mathcal{N}(A_Q^T)$. Below we refer to both kernel steps in the Kernel-Step algorithm and kernel-like steps in the Regularized algorithm as “kernel-like steps”. We have found that $\alpha = \sqrt{\bar{\delta}}$ counts the kernel-like steps in an appropriate manner.

The observed behavior is that both algorithms take a series of consecutive kernel-like steps, during which the objective is increased very rapidly while `termcrit` remains relatively constant, and then switch over to regular-steps which rapidly decrease `termcrit` to tolerance. In general, on this problem class, the Kernel-

Step algorithm takes almost exactly k kernel-like steps, where k is the tube-dimension, while the regularized algorithm takes at least k and often almost exactly $u \times k$ where u is an integer that depends on the size of the bound constraints (the cube). Intuitively, what is happening, is that the true kernel steps make it all the way down the tube, while the Regularized algorithm’s kernel-like steps only make it all the way for tubes that are not too long, or otherwise only makes it halfway, or a third of the way, etc.

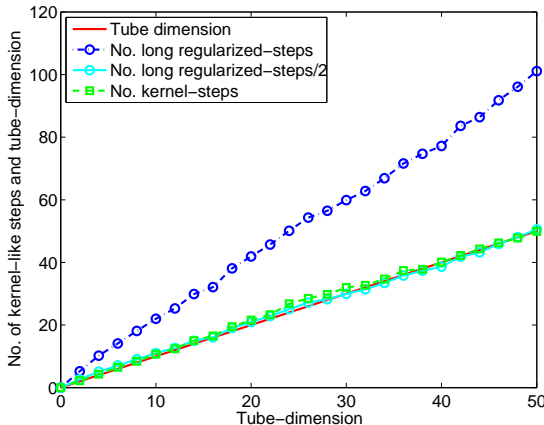


Figure 3: Tube-in-cube problem of size 50×2600 , with “cube” constraint of the form $\|y\|_\infty \leq R$, with $R = 100$, and with varying degrees of rank-degeneracy, namely, with tube-dimension $k = m - r$ (r the rank of A_{tube}) ranging between 0 (non-degenerate) and 45 (highly degenerate). The plot shows the number of kernel steps vs. the tube-dimension for the Kernel-Step algorithm, as well as the number kernel-like steps for the Regularized algorithm. For both algorithms we used the Most Active rule with $M = 3m$. We counted a kernel-like step whenever (60) was satisfied. Examining the results, we see that the tube-dimension is a very good predictor of the number of kernel-like steps that will be taken on the problem. The observed behavior is that both algorithms take about k or $2k$ consecutive kernel-like steps for the Kernel-Step and Regularized algorithm, respectively, where k is the tube-dimension, during which the objective is increased rapidly while `termcrit` remains relatively constant, and then switch over to regular steps which rapidly decrease `termcrit` to tolerance.

Figure 4 shows a comparison of the performance of the two algorithms. The figures show the time and iteration count vs. the fraction of constraint set kept in Q at each iteration (using the Most-Active rule to select Q) on some test problems. For the test problems, we generated 10 random tube-in-cube problems with $m = 100$, $n = 10000$, and with $\dim \mathcal{N}(A_{\text{tube}}^T) = 50$, and “cube” constraint of the form $\|y\|_\infty \leq R$, with $R = 100$. The Kernel-Step algorithm outperformed the Regularized algorithm on this class of problems in both iterations and time. Qualitatively, the Kernel-Step method uses kernel steps almost exclusively at the beginning and get the iterate quickly to a “non-degenerate” corner of the feasible set, while the Regularized algorithm bounces around a bit longer in the tube.

5.3 Random sparse problems

As discussed in the introduction, the sparsity of a constraint matrix A matrix can be correlated with the degree of rank-degeneracy that the corresponding problem will exhibit. To make this statement concrete, we need a way to quantify the “degree of rank-degeneracy” of a given linear program. Toward this end,

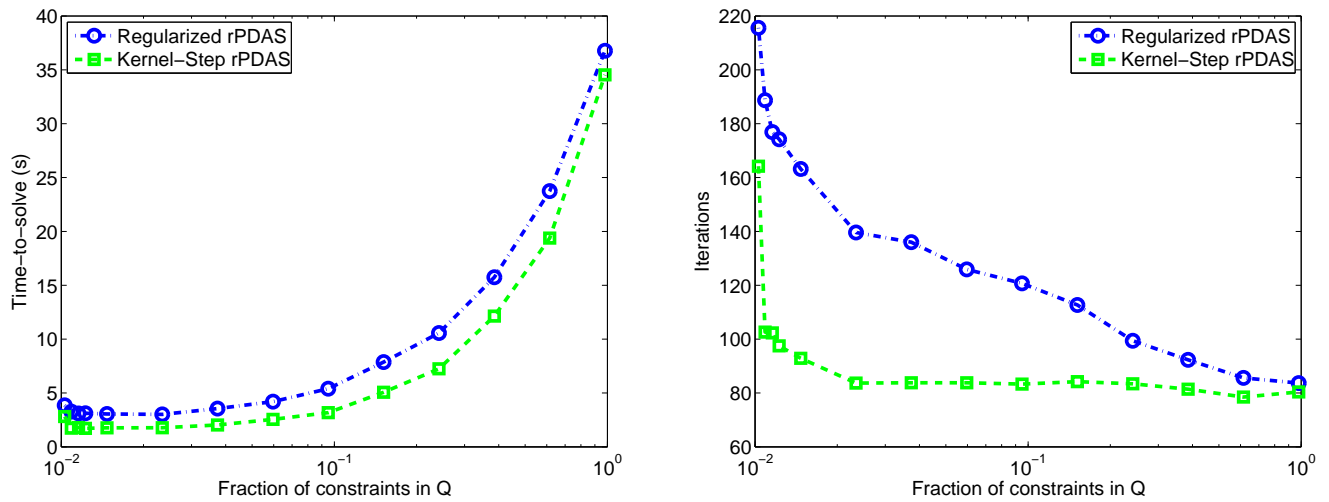


Figure 4: Performance of the Regularized and Kernel-Step rPDAS methods on the tube-in-cube using the Most Active rule. The plots show average total time (in seconds) and iteration count to complete solution, vs. the fraction of the total constraints used at each iteration for 10 instances of the tube-in-cube problem with $m = 100$, $n = 10000$, tube-dimension $k = 50$, and “cube” constraint of the form $\|y\|_\infty \leq R$, with $R = 100$. The horizontal axis shows the size $|Q|$ of the constraint set (fixed through each entire optimization run) as a fraction of n .

we define $d(A)$ to be the average value of $\dim \mathcal{N}(A_Q^T)$ over all possible subsets $Q \in \mathbf{n}$ with $|Q| = m$.⁶(To compare the degree of rank-degeneracy of linear programs of differing size we can normalize this quantity by m .) This quantity very nearly reproduces the “tube dimension” for tube-in-cube type problems and shows strong positive correlation with the level of sparsity of A as we demonstrate below.

We generated a sequence of random problems as in section 5.2, except that, in the generation of A , we specified the sparsity (or fraction of non-zeros), rather than the explicit tube-dimension. We appended bound constraints of the form $\|y\|_\infty \leq R$, again with $R = 100$, to ensure $\text{rank}(A) = m$.

Figure 5 plots an estimate $\hat{d}(A)$ of the above-defined measure of rank-degeneracy $d(A)$ (which we think of as an estimate of the “tube-dimension” for this sparse class of problems) versus the percentage of non-zeros in A , averaged over 10 problems of size 100×10000 , at each sparsity level. To estimate $d(A)$ we sample $N = 10$ sets Q^k , with $|Q^k| = m$, $k \in \{1, 2, \dots, N\}$ at random from the columns of A and compute $\hat{d}(A) = \frac{1}{N} \sum_{k=1}^N (m - \text{rank}(A_{Q^k}))$, where $\text{rank}(A_{Q^k})$ is computed as the number of singular values of A larger than a threshold which we took to be $n \times \text{eps}(\|A\|)$, which is the Matlab default tolerance for rank estimation (`eps` is a built-in Matlab function). Also shown in the figure is the average number of kernel-like steps taken by the two algorithms on these problems. We see that there is indeed a strong correlation between rank-degeneracy and sparsity and, as was the case with the tube-in-cube problem, the number of kernel steps that will be taken can be predicted fairly accurately by the estimated “tube-dimension” $\hat{d}(A)$ of the problem.

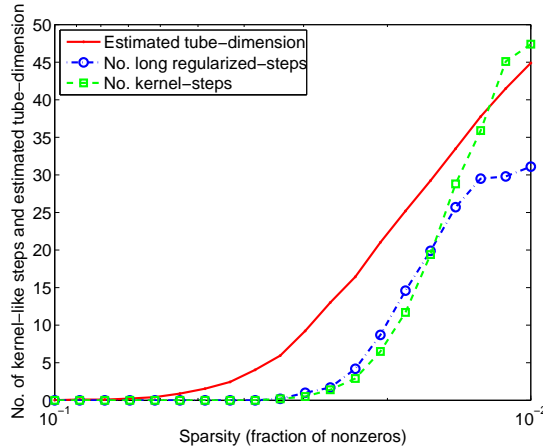


Figure 5: Estimated tube-dimension $\hat{d}(A)$ versus percentage of non-zeros in A . Also shown is the number of kernel-like steps taken by the Kernel-Step and Regularized rPDAS algorithms vs. the fraction of non-zeros in the sparse constraint matrix A , of a randomly generated sparse LP. (For both algorithms we used the Most-Active rule with $M = 3m$.) Down to about 1% non-zeros (note the fraction of non-zeros decreases, so sparsity increases, toward the right), the increasing sparsity is accompanied by increasing rank-degeneracy, and the number of kernel-like steps is roughly given by the “tube-dimension”.

Next, we compared the performance of the algorithms on the random sparse problem. As was done for the tube-in-cube problems, we ran the two algorithms on 10 randomly generated sparse problems, and

⁶More generally, we could define a parametrized measure $d(A, M)$ as the average of $\dim \mathcal{N}(A_Q^T)$ over all $Q \in \mathbf{n}$ with $|Q| = M$, where notionally M might be set closer to the average size of Q used in the algorithms. After some experimentation, we found the measure to be most accurate in estimating the “tube-dimension” and counts of kernel-like steps when M was set equal to m , and thus did so and dropped the parameter from the definition. The more general version may still be useful for future work.

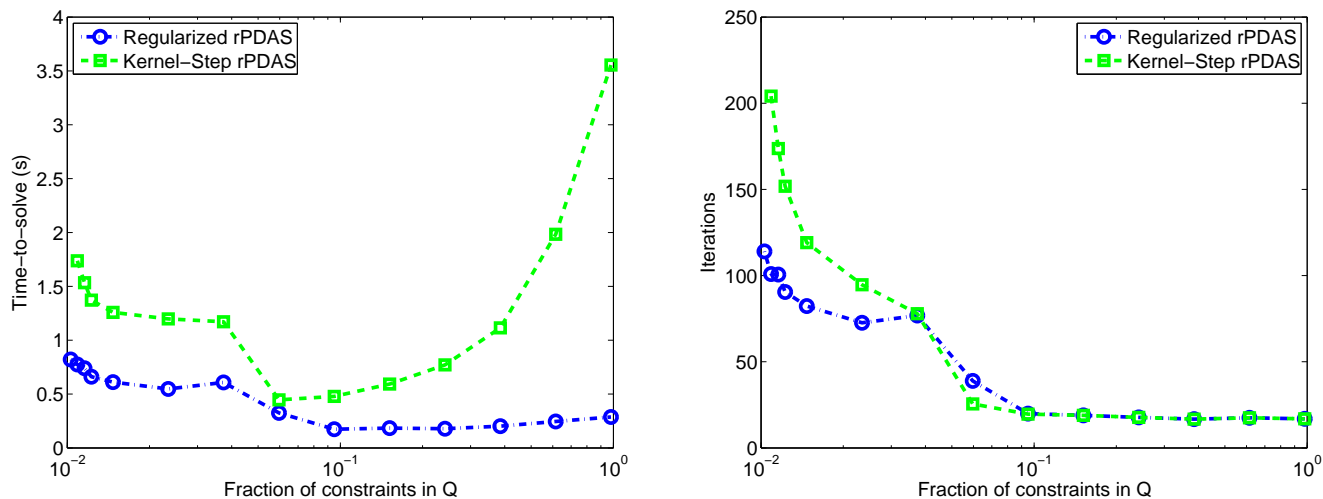


Figure 6: Performance of the regularized and kernel-step rPDAS methods the sparse random problems using the Most Active rule. The plots show average total time (in seconds) to solve and iteration count vs. the fraction of the total constraints used at each iteration for 10 instances of the sparse random problem with $m = 100$, $n = 10000$. In each case the constraint matrix A was generated with 1% non-zeros. The horizontal axis shows the size $|Q|$ of the constraint set used at each iteration of the specific run, as a fraction of n .

recorded the time and iteration count to solve the problem to tolerance for varying degrees of constraint reduction. The results are shown in Figure 6.⁷ Note that, for the Kernel-Step algorithm, the time-to-solve increases rapidly with increasing size of Q ; see Remark 2.

5.4 Further test problems

Finally, to assess the value of the regularization/kernel-step concepts, we tested Algorithms 1 and 2 on a pair of sparse, highly imbalanced, real world problems from the netlib [net] collection, rail507 ($m = 507$, $n = 63516$) and rail582 ($m = 582$, $n = 56097$), and compared them to an unregularized version of rPDAS (with $\bar{\delta} = 0$), with the policy that if the selected constraint set resulted in $\text{rank}A_Q < m$, the working constraint set size would be doubled (still selecting constraints in order of smaller s_i first). These two problems were also among those tested in [WNT012], where the constraint-reduced algorithm developed in that paper was seen to outperform several prior constraint-reduction or “build-up” algorithms, which generally outperformed their corresponding “unreduced” variants.⁸ For Regularized rPDAS, we set $\bar{\delta} = 10^{-5}$ and, in all cases, we started the iteration from $(y_0, s_0 = c - A^T y_0)$, an approximate analytic center of the dual feasible set

⁷On that figure, with Regularized rPDAS, it seems that the total time is almost independent of $|Q|$ when $|Q|/n$ is larger than about 0.1. This is because these problems are rather small, and the time is dominated by overhead.

⁸For these two problems, to overcome observed numerical problems near the solution, the algorithms in [WNT012] used an *ad-hoc*, *fixed* regularization of the normal equations, as in (16) with $\delta = 10^{-5}$. Furthermore, when numerical difficulties prevented factorization of the normal matrix even when the working set was increased to $Q = \mathbf{n}$, following [TAW06], the algorithm would fall back on Matlab’s `cholinc(·, ‘inf’)`. This problem was observed in the current investigation as well, but here we fall back on the symmetric-indefinite factorization of the augmented system (14), using Matlab’s command `ldl`, rather than rely on `cholinc`.

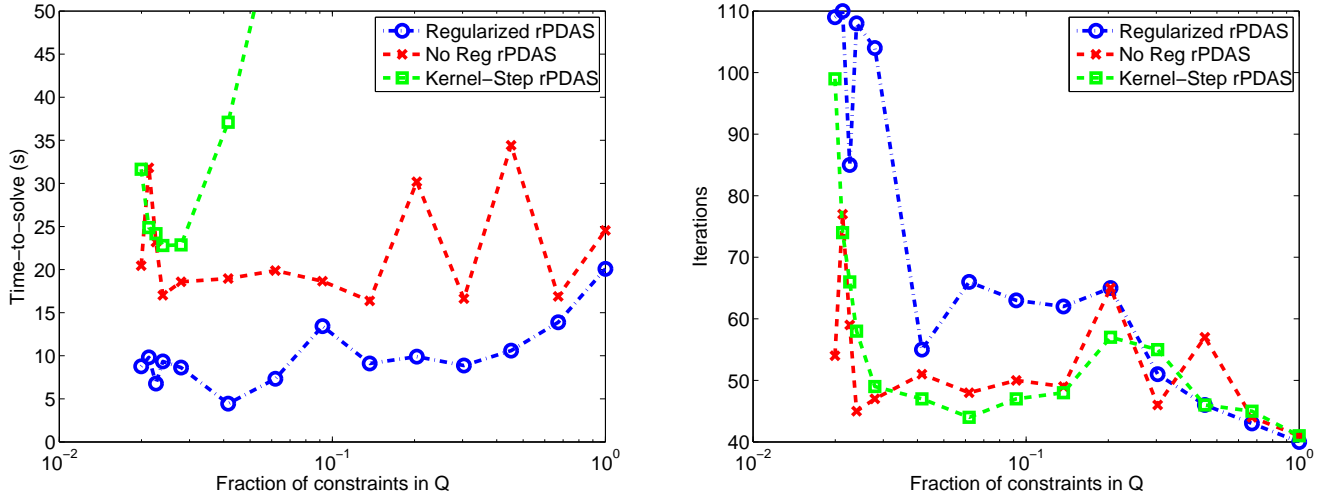


Figure 7: Performance of the regularized and unregularized rPDAS methods on the rail507 problem using the Most Active rule. The plots show average total time (in seconds) to solve and iteration count vs. the size $|Q|$ of the constraint set used at each iteration of the specific run, as a fraction of n .

generated using third party software, and $x_0 := e$, the vector of ones. The results are shown in Figures 7 and 8, which show the running time and iteration count as a function of $|Q|$, again as a fraction of the total set. For both problems, Regularized and Kernel-Step rPDAS algorithms have similar iteration-counts as the corresponding unregularized method, but Regularized rPDAS has significantly lower running time. This is because the unregularized algorithm generally goes through several “doubling” steps before obtaining a full rank A_Q . The iteration counts for Kernel-Step rPDAS compare favorably to the others, but much more computation time is required. This is because, as discussed in Remark 2 above, the QR factorization used in Kernel-Step rPDAS is computationally costly compared to the operations used in the regularized and unregularized routines.

6 Concluding Remarks

Two constraint-reduced primal-dual interior-point schemes were proposed that address the oft-encountered situation—which we term “rank-degeneracy”—where the “reduced” A matrix A_Q (with only a small subset of $|Q| > m$ columns of the original $m \times n$ matrix) is rank-deficient. The first is based on a type of regularization, and the second is based on a new type of search direction called the “kernel” step, which can be viewed as a limiting case of the first. Both schemes were analyzed and numerically tested in the context of rPDAS, a previously proposed dual-feasible constraint-reduced primal-dual affine-scaling algorithm. Global and quadratic local convergence were proved under mild assumptions, and promising numerical results were obtained with both. As for the respective practical merit of the two algorithms, according to our tests, the Regularized rPDAS appears to have the edge on highly sparse problems, while the Kernel-Step rPDAS seems

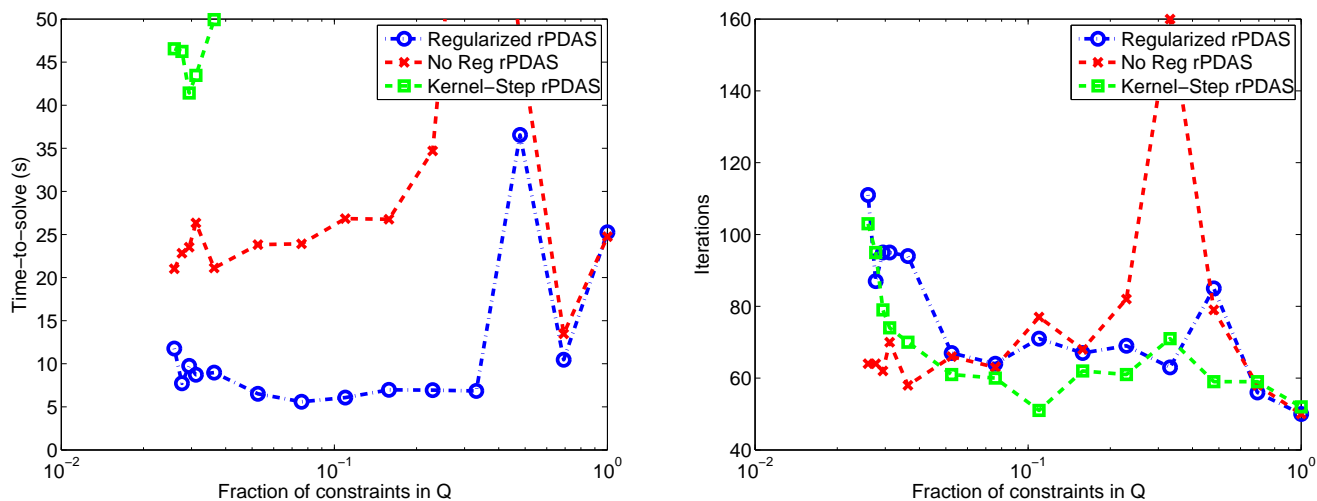


Figure 8: Performance of the regularized and unregularized rPDAS methods on the rail582 problem using the Most Active rule. The plots show average total time (in seconds) to solve and iteration count vs. the size $|Q|$ of the constraint set used at each iteration of the specific run, as a fraction of n .

to be faster in the case of the dense, highly rank-degenerate “tube-in-cube” problems. Finally, extensions to predictor-corrector variants and to “infeasible” algorithms were mentioned.

Acknowledgments

The authors wish to thank Dianne O’Leary for helpful discussions, and Ming-Tse Paul Laiu for running some numerical comparisons.

References

- [BT97] D. Bertsimas and J. Tsitsiklis, *Introduction to linear optimization*, Athena, 1997.
- [Dik74] I.I. Dikin, *On convergence of an iterative process*, *Upravlyaemye Systemy* **12** (1974), 54–60, In Russian.
- [DY91] G. Dantzig and Y. Ye, *A build-up interior-point method for linear programming: Affine scaling form*, Working paper, Department of Management Science, University of Iowa, 1991.
- [FO07] M. P. Friedlander and D. Orban, *Exact primal-dual regularization of linear programs*, 2007, Presentation given at ICCOPT: Hamilton, Ontario.
- [FO12] M.P. Friedlander and D. Orban, *A primal-dual regularized interior-point method for convex quadratic programming*, *Math. Programming Computation* **4** (2012), no. 1, 71–107.
- [HT12] Meiyun Y. He and André L. Tits, *Infeasible constraint-reduced interior-point methods for linear optimization*, *Optimization Methods and Software* **27** (2012), no. 4–5, 801–825.
- [Meh92] S. Mehrotra, *On the implementation of a primal-dual interior point method*, *SIAM Journal on Optimization* **2** (1992), no. 4, 575–601.

- [net] *Netlib linear programming test problems*, <http://www-fp.mcs.anl.gov/OTC/Guide/TestProblems/LPtest/>.
- [Sai96] R. Saigal, *A simple proof of a primal affine scaling method*, *Annals of Operations Research* **62** (1996), no. 1, 303–324.
- [ST96] M. A. Saunders and J. A. Tomlin, *Solving regularized linear programs using barrier methods and KKT systems*, Tech. report, Stanford University, Department of EES, 1996.
- [Ste89] G. W. Stewart, *On scaled projections and pseudo-inverses*, *Linear Algebra and its Applications* **112** (1989), 189–194.
- [TAO06] A. L. Tits, P. A. Absil, and D. P. O’Leary, *Constraint reduction for certain degenerate linear programs*, July 30-August 4 2006, presented at 19th ISMP.
- [TAW06] A.L. Tits, P.A. Absil, and W. Woessner, *Constraint reduction for linear programs with many constraints*, *SIAM Journal on Optimization* **17** (2006), no. 1, 119–146.
- [Win10] Luke Winternitz, *Primal-dual interior point algorithms for linear programs with many inequality constraints*, Ph.D. thesis, University of Maryland, 2010.
- [WNT012] L. B. Winternitz, S. O. Nicholls, A. L. Tits, and D. P. O’Leary, *A constraint reduced variant of Mehrotra’s predictor-corrector algorithm*, *Computational Optimization and Applications* **51** (2012), no. 1, 1001–1036.
- [Wri97] S. J. Wright, *Primal-dual interior-point methods*, SIAM, Philadelphia, 1997.