Metric Selection in Fast Dual Forward Backward Splitting ⋆

Pontus Giselsson * **Stephen Boyd** *

* *Electrical Engineering Department, Stanford University.*
Email: {pontusg,boyd}@stanford.edu

Abstract:
The performance of fast forward-backward splitting, or equivalently fast proximal gradient methods, is susceptible to conditioning of the optimization problem data. This conditioning is related to a metric that is defined by the space on which the optimization problem is stated. By selecting a space on which the optimization data is more well-conditioned, the performance of the fast forward-backward splitting method is likely to improve. In this paper, we propose several methods, with different computational complexity, to find a space on which the algorithm performs well when applied to the dual of a strongly convex optimization problem. We evaluate the proposed metric selection procedures by comparing the performance to when the fast dual forward-backward splitting method is applied in the Euclidean space. For the most ill-conditioned problem we consider, the computational complexity is improved by two to three orders of magnitude. We also report compatible to superior performance compared to state-of-the-art optimization software.

1. INTRODUCTION

Fast gradient methods have been around since the early 80's when the seminal paper Nesterov (1983) was published. The fast gradient method in Nesterov (1983) is applicable to unconstrained smooth optimization problems. This method was not well recognized until the mid 00's, after which several extensions and generalizations of the fast gradient method have been proposed. In Nesterov (2003), new acceleration schemes were presented as well as methods for constrained optimization, i.e., projected fast gradient methods. In Nesterov (2005), smoothing techniques for the fast gradient method for nonsmooth problems were presented. Fast proximal gradient methods, or equivalently a fast forward-backward splitting methods, that solve problems of the form

$$\text{minimize} \quad f(x) + g(x) \tag{1}$$

where f is convex and smooth and g is a proper, closed, and possibly extended-valued convex function, were presented in Nesterov (2013); Beck and Teboulle (2009). In Tseng (2008), generalizations and unifications of many fast forward-backward splitting methods were presented.

The smooth part of the composite objective function, f in (1), is in fast forward-backward splitting approximated by the r.h.s. of

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \tfrac{\beta}{2}\|x - y\|^2 \tag{2}$$

where the norm and inner-product are given by the space on which f is defined. The condition that (2) holds for all x and y is referred to as f being β-smooth w.r.t. the space on which f is defined. Since the r.h.s. of the smoothness condition (2) is the only information the algorithm has about the smooth function, the smaller the gap in (2) (i.e. the better the r.h.s. of (2) approximates f), the better the performance of the algorithm is likely to be. In this paper, we show how to select a space (or metric, we will use these notions interchangeably since the metric defines the space) on which the fast forward-backward splitting method performs well, when solving the dual of strongly convex composite optimization problems.

The spaces we consider when solving these problems are Euclidean spaces with inner product $\langle x, y \rangle = x^T y$ and a scaled norm $\|x\|_K = \sqrt{x^T K x}$, where K is a positive definite metric matrix that defines how distances are measured. In this paper, we analyze how K should be chosen to get efficient fast proximal gradient methods when solving the dual of strongly convex optimization problems. This is done by providing tight quadratic majorizers to the smooth part of the dual problem, and by showing how this knowledge can be utilized to improve the convergence of the algorithms. We also provide a number of different computational methods, with different computational complexity, that find the desired metric matrix K exactly or approximately.

Recently, Richter et al. (2013); Patrinos and Bemporad (2014) proposed fast dual forward-backward splitting as appropriate optimization algorithms for embedded model predictive control. They propose two different dual formulations and apply fast forward-backward splitting in the Euclidean space on their respective dual problems. Using the results in this paper, we show how to select an appropriate space on which to apply these methods. The perfor-

⋆ During the preparation of this paper, the first author was a member of the LCCC Linnaeus Center at Lund University. Financial support from the Swedish Research Council for the first author's Postdoctoral studies at Stanford University is gratefully acknowledged. Further, Alexander Domahidi is gratefully acknowledged for suggesting the AFTI-16 control problem as a benchmark and providing FORCES code for the same. Finally, the authors acknowledge the anonymous reviewers for constructive feedback.

mance improvement is evaluated by applying the methods to a pitch control problem in an AFTI-16 aircraft that has previously been studied in Kapasouris et al. (1990); Bemporad et al. (1997). This is a challenging problem for first order methods since it is fairly ill-conditioned. The numerical evaluation shows that for such ill-conditioned problems, the computation time can be improved by two to three orders of magnitude by selecting the space on which to apply the algorithm appropriately. We also compare the performance to the algorithm in O'Donoghue et al. (2013); Jerez et al. (2013) that use an ADMM-based (see Boyd et al. (2011) for more on ADMM - the alternating direction method of multipliers) algorithm to solve the embedded model predictive control optimization problems. Our algorithms are implemented in the MATLAB toolbox QPgen Giselsson (2014a). We compare the performance of QPgen to DuQuad, see Necoara and Patrascu (2015), which is based on the dual gradient methods in Nedelcu et al. (2014); Necoara and Nedelcu (2014) that use inexact inner minimizations instead of splitting methods as in Richter et al. (2013); Patrinos and Bemporad (2014) and here. We further compare to FiOrdOs, see Ullmann and Richter (2012), which implements, e.g., the method in Richter et al. (2013) and the method in Chambolle and Pock (2011). We compare to FORCES Domahidi et al. (2012) and CVXGEN Mattingley and Boyd (2012), that are C code-generators that use tailored interior point methods. We compare to qpOASES, which implements the online active set method in Ferreau et al. (2008). Further, we compare to the MPTtoolbox Herceg et al. (2013), which is based on the explicit MPC approach in Bemporad et al. (2002). Finally, we compare to the general commercial QP-solver MOSEK, see ApS (2013). The proposed fast dual forward-backward splitting when applied on a suitable space performs little to much better than all other methods on the considered example.

Fast dual forward-backward splitting can also be used for distributed optimization when the objective to be minimized is separable. In the context of gradient methods, this has been known since Everett (1963); Danzig and Wolfe (1961); Benders (1962). Recently such approaches have been proposed for distributed model predictive control (DMPC) Negenborn (2007); Doan et al. (2011); Giselsson et al. (2013); Giselsson (2013), and resource optimization over networks Ghadimi et al. (2013); Beck et al. (2013); Necoara and Nedelcu (2015). Often, centralized coordination is needed when selecting the step-size for the gradient-step. In Beck et al. (2013), it was noted that the smooth part of the dual problem consists of a sum of local dual terms. Each of these can compute its own step-size, share with its neighbors, and sum to get a fully distributed step-size selection. This procedure can be augmented by the results of this paper to not only select a local step-size, but local matrices K_i that, after neighborhood exchange and summation, give a space on which to apply the fast distributed dual forward-backward splitting. When solving a set of randomly generated separable optimization problems, the iteration count of the algorithm when applied on a space as computed by the methods presented in this paper is two to ten times lower than the iteration count when the local step-sizes are computed as in Beck et al. (2013). We also compare the performance to the recently proposed dual Newton conjugate gradient method in Kozma et al. (2014). The proposed methods outperform this method by at least a factor of 10 on the considered examples.

This paper unifies and extends the conference publications Giselsson (2014b,c); Giselsson and Boyd (2014).

2. NOTATION AND PRELIMINARIES

2.1 Notation

We denote by $\mathbb{R}$, $\mathbb{R}^n$, $\mathbb{R}^{m\times n}$, the sets of real numbers, column vectors, and matrices. We use notation $(x, y, z) := [x^T\ y^T\ z^T]^T$ for stacked real column vectors. We also use notation $\overline{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$ for the extended real line. $\mathbb{S}^n \subseteq \mathbb{R}^{n\times n}$ is the set of symmetric matrices, and $\mathbb{S}^n_{++} \subseteq \mathbb{S}^n$, $[\mathbb{S}^n_+] \subseteq \mathbb{S}^n$, are the sets of positive [semi] definite matrices. We consider real finite-dimensional normed vector spaces with the Euclidean inner product $\langle x, y\rangle = x^T y$ and different norms. When using the induced norm $\|x\| = \sqrt{\langle x, x\rangle}$, we get the standard Euclidean space. We also consider spaces $\mathbb{E}_H$ (with Euclidean inner product) with scaled norm $\|x\|_H = \sqrt{\langle x, Hx\rangle}$, where $H \in \mathbb{S}^n_{++}$. The first element x in the inner product $\langle x, y\rangle$ on $\mathbb{E}_H$ is from the dual space, denoted by $\mathbb{E}^*_H$, i.e., $x \in \mathbb{E}^*_H$, while the second element $y \in \mathbb{E}_H$. The dual norm to $\|y\|_H$ is $\|y\|^*_H = \max_x \{\langle y, x\rangle_2 : \|x\|_H = 1\} = \|y\|_{H^{-1}}$, i.e., $\mathbb{E}^*_H = \mathbb{E}_{H^{-1}}$. Further, the class of closed, proper, and convex functions $f : \mathbb{E}_H \to \overline{\mathbb{R}}$ is denoted by $\Gamma_0(\mathbb{E}_H)$. The conjugate function $f^* : \mathbb{E}^*_H \to \overline{\mathbb{R}}$ to $f \in \Gamma_0(\mathbb{E}_H)$ is defined as $f^*(y) = \sup_x \{\langle y, x\rangle - f(x)\}$. The adjoint operator to a bounded linear operator $\mathcal{A} : \mathbb{E}_H \to \mathbb{E}_K$ is denoted by $\mathcal{A}^* : \mathbb{E}^*_K \to \mathbb{E}^*_H$ and is defined as the unique operator that satisfies $\langle \mathcal{A}x, y\rangle = \langle \mathcal{A}^* y, x\rangle$ for all $x \in \mathbb{E}_H$ and $y \in \mathbb{E}^*_K$. Since the ambient space for $\mathbb{E}_H$ is the standard Euclidean space, we often denote the matrix that corresponds to the operator $\mathcal{A} : \mathbb{E}_H \to \mathbb{E}_K$ by $A \in \mathbb{R}^{m\times n}$. Finally, $I_{\mathcal{X}}$ denotes the indicator function for the set $\mathcal{X}$, and $I_{g(x)\leq 0}$ denotes the indicator function for the set $\mathcal{X} = \{x \mid g(x) \leq 0\}$.

2.2 Preliminaries

In this section, we introduce the concepts of strong convexity and strong smoothness and present a result on how these concepts relate to each other through the conjugate function.

Definition 1. A function $f \in \Gamma_0(\mathbb{E}_H)$ is *β-strongly convex* (w.r.t. $\mathbb{E}_H$) if $f - \frac{\beta}{2}\|\cdot\|^2_H$ is convex.

Definition 2. A function $f \in \Gamma_0(\mathbb{E}_H)$ is *β-smooth* (w.r.t. $\mathbb{E}_H$) if it is differentiable and $\frac{\beta}{2}\|\cdot\|^2_H - f$ is convex.

Remark 3. An equivalent characterization of β-smoothness w.r.t. $\mathbb{E}_H$ is that

$$f(x) \leq f(y) + \langle \nabla f(y), x - y\rangle + \tfrac{\beta}{2}\|x - y\|^2_H \qquad (3)$$

holds for all $x, y \in \mathbb{E}_H$.

Next, we state a result that links the notions of strong convexity and smoothness though the conjugate function. A proof to this, in a more general setting, can be found, e.g., in (Zalinescu, 2002, Proposition 3.5.3).

Proposition 4. Suppose that $f \in \Gamma_0(\mathbb{E}_H)$. Then the following are equivalent:

(i) f is β-strongly convex (w.r.t. $\mathbb{E}_H$)
(ii) f^* is $\frac{1}{\beta}$-smooth (w.r.t. $\mathbb{E}_H^* = \mathbb{E}_{H^{-1}}$)

Remark 5. The converse statement also holds since $f = (f^*)^*$ for $f \in \Gamma_0(\mathbb{E}_H)$.

3. PROBLEM FORMULATION

We consider optimization problems of the form

$$\begin{array}{ll} \text{minimize} & f(x) + g(y) \\ \text{subject to} & \mathcal{A}x = y \end{array} \tag{4}$$

We assume that the following assumption holds throughout the paper:

Assumption 6.

(a) The extended valued function $f \in \Gamma_0(\mathbb{E}_H)$ is 1-strongly convex (w.r.t. $\mathbb{E}_H$) with $H \in \mathbb{S}^n_{++}$.
(b) The extended valued function $g \in \Gamma_0(\mathbb{E}_K)$ with $K \in \mathbb{S}^m_{++}$.
(c) $\mathcal{A} : \mathbb{E}_H \to \mathbb{E}_K$ is a bounded linear operator.

Remark 7. A function that satisfies Assumption 6(a) is $f(x) = \frac{1}{2}x^T Hx + \hat{f}$ where $H \in \mathbb{S}^n_{++}$ and $\hat{f} \in \Gamma_0(\mathbb{E}_H)$.

Since g (and $\hat{f}$) are allowed to be extended valued, they can, e.g., be indicator functions for nonempty, closed, and convex constraint sets.

The operator $\mathcal{A} : \mathbb{E}_H \to \mathbb{E}_K$ has an associated matrix $A : \mathbb{R}^n \to \mathbb{R}^m$ that satisfies $\mathcal{A}x = Ax$ for all $x \in \mathbb{R}^n$.

To arrive at the dual problem, we introduce Lagrange multipliers $\mu \in \mathbb{E}_{K^{-1}}$, to get Lagrangian

$$L(x, y, \mu) = f(x) + g(y) + \langle \mathcal{A}x - y, \mu \rangle.$$

By minimizing the Lagrangian over x, and y, we get

$$\begin{aligned} &\inf_{x,y} L(x, y, \mu) = \inf_x \{\langle \mathcal{A}^*\mu, x\rangle + f(x)\} + \inf_y \{\langle -y, \mu\rangle + g(y)\} \\ &= -\sup_x \{\langle -\mathcal{A}^*\mu, x\rangle - f(x)\} - \sup_y \{\langle \mu, y\rangle - g(y)\} \\ &= -f^*(-\mathcal{A}^*\mu) - g^*(\mu). \end{aligned}$$

Negating this, we get the negated dual problem to (4) (see, e.g, (Rockafellar, 1970 §31) for more details):

$$\text{minimize } d(\mu) + g^*(\mu) \tag{5}$$

where

$$d(\mu) := f^*(-\mathcal{A}^*\mu). \tag{6}$$

Note that $d, g^* \in \Gamma_0(\mathbb{E}_{K^{-1}})$.

The performance of fast dual forward backward splitting is highly sensitive to on which space the algorithm is applied. This paper is about computing metrics H and K, based on problem data, that define spaces on which fast dual forward backward splitting performs well.

4. DUAL PROBLEM PROPERTIES

In this section, we will present tight characterizations of the smooth part of the dual problem. These characterizations will later guide us in choosing space/metric on which to apply the forward-backward splitting algorithm.

From, e.g., (Rockafellar, 1970, Theorem 23.5) combined with the chain rule, we know that d is differentiable with

$$\nabla d(\mu) = -\mathcal{A}^* x^\star(\mu) \tag{7}$$

where

$$x^\star(\mu) = \operatorname*{argmin}_x \{\langle \mathcal{A}^*\mu, x\rangle + f(x)\}. \tag{8}$$

It is also well known, see e.g. (Nesterov, 2005, Theorem 1), that ∇d is Lipschitz continuous with constant $\|\mathcal{A}^*\|^2 = \|\mathcal{A}\|^2$ (since f is 1-strongly convex (w.r.t. $\mathbb{E}_H$) due to Assumption 6), where the norm is the operator norm. (This also follows directly from Proposition 4, i.e., from (Zalinescu, 2002, Proposition 3.5.3), and the Cauchy-Schwarz inequality.) Since $\mathcal{A} : \mathbb{E}_H \to \mathbb{E}_K$, the norm $\|\mathcal{A}\|$ is defined by

$$\begin{aligned} \|\mathcal{A}\| &= \max_x \{\|\mathcal{A}x\|_K \ : \ \|x\|_H \le 1\} \\ &= \left\{\|K^{1/2}Ax\|_2 \ : \ \|H^{1/2}x\|_2 \le 1\right\} \\ &= \left\{\|K^{1/2}AH^{-1/2}v\|_2 \ : \ \|v\|_2 \le 1\right\} \\ &= \|K^{1/2}AH^{-1/2}\|_2 \end{aligned}$$

where $\|\cdot\|_2$ denotes the standard induced Euclidean norm and A is the Euclidean matrix representation of $\mathcal{A}$. By defining d on $\mathbb{E}_I$, i.e., by choosing $K = I$, we get that $d \in \Gamma_0(\mathbb{E}_I)$ is Lipschitz continuous with constant $\|AH^{-1/2}\|_2^2 = \|AH^{-1}A^T\|_2$. This is exactly the Lipschitz constant provided in Richter et al. (2013) which implies that

$$d(\mu) \le d(\nu) + \langle \nabla d(\nu), \mu - \mu\rangle_2 + \tfrac{\|AH^{-1}A^T\|_2}{2}\|\mu - \nu\|_2^2 \tag{9}$$

holds for all $\mu, \nu \in \mathbb{R}^m$. This upper bound can be improved by defining d on $\mathbb{E}_{K^{-1}}$ with $K = (AH^{-1}A^T)^{-1}$ (where we have implicitly assumed that A has full row rank). This implies that d is 1-smooth w.r.t. $\mathbb{E}_{AH^{-1}A^T}$, i.e., that

$$d(\mu) \le d(\nu) + \langle \nabla d(\nu), \mu - \mu\rangle_2 + \tfrac{1}{2}\|\mu - \nu\|^2_{AH^{-1}A^T} \tag{10}$$

holds for all $\mu, \nu \in \mathbb{E}_{AH^{-1}A^T}$. This is obviously a tighter characterization of d than (9).

Remark 8. We improve the upper bound on d by defining it on a different space. It is straight-forward to verify that this does not influence the shape of the function d itself, only the bound is improved.

When selecting $K = (AH^{-1}A^T)^{-1}$, we implicitly assume that A has full row rank. In the following result, we show that (10) also holds when A is not full row rank.

Proposition 9. Suppose that Assumption 6 holds. Then $d \in \Gamma_0(\mathbb{E}_{K^{-1}})$ as defined in (6) satisfies

$$d(\mu) \le d(\nu) + \langle \nabla d(\nu), \mu - \nu\rangle + \tfrac{1}{2}\|\mu - \nu\|_L^2 \tag{11}$$

for any $L \succeq AH^{-1}A^T$ and for all $\mu, \nu \in \mathbb{E}_{K^{-1}}$, where $A \in \mathbb{R}^{m\times n}$ is the matrix representation of $\mathcal{A}$.

Proof. Since Assumption 6 states that f is 1-strongly convex w.r.t. $\mathbb{E}_H$, Proposition 4, gives that f^* is 1-smooth w.r.t. $\mathbb{E}_{H^{-1}}$. Thus, (3) holds for f^* for any $x, y \in \mathbb{E}_{H^{-1}}$. Further, since f^* is independent of the norm on the space (it only depends on the inner product) (3) holds for f^* also for any $x, y \in \mathbb{R}^n$. Especially, let $x = -A^T\mu$ and $y = -A^T\nu$ to get

$$\begin{aligned} &d(\mu) = f^*(-A^T\mu) \\ &\le d(\nu) + \langle \nabla f^*(-A^T\nu), -A^T(\mu - \nu)\rangle + \tfrac{1}{2}\|A^T(\mu - \nu)\|^2_{H^{-1}} \\ &= d(\nu) + \langle -A\nabla f^*(-A^T\nu), \mu - \nu\rangle + \tfrac{1}{2}\|\mu - \nu\|^2_{AH^{-1}A^T} \\ &= d(\nu) + \langle \nabla d(\nu), \mu - \nu\rangle + \tfrac{1}{2}\|\mu - \nu\|^2_{AH^{-1}A^T}. \end{aligned}$$

Since the the inequality holds for $AH^{-1}A^T$, it also holds for any $L \succeq AH^{-1}A^T$. This concludes the proof.

Next, we show that for many interesting functions f, the bound provided in Proposition 9 on d is indeed tight. Essentially, we show that if the strong convexity bound on the primal (i.e., the quadratic lower bound to the primal) is tight, so is the smoothness bound on the dual (i.e. the quadratic upper bound to the dual). A proof is provided in Appendix A.

Proposition 10. Suppose that Assumption 6 holds and that there exists a full-dimensional ball $\mathcal{B}_r^n(x^\star(\bar{\mu}))$ centered around $x^\star(\bar{\mu})$ for some $\bar{\mu} \in \mathbb{E}_{K^{-1}}$ on which $f - \frac{1}{2}\|\cdot\|_H^2$ is linear. Then no matrix $L \not\succeq AH^{-1}A^T$ exists such that $d : \mathbb{E}_{K^{-1}} \to \mathbb{R}$ as defined in (6) satisfies (11) for all $\mu, \nu \in \mathbb{E}_{K^{-1}}$.

The assumptions in Proposition 10 are met, for instance, if A has full column rank and $f(x) = \frac{1}{2}\|x\|_H^2 + h(x)$ where h is the indicator function for a closed and convex constraint set with nonempty interior, the 1-norm, a linear function, or any other function that is linear on a convex subset with nonempty interior. For these examples, Proposition 10 indeed gives the best obtainable quadratic majorizer of d. However, for f being a quadratic plus the indicator functions for an affine subspace $I_{Bx=b}$, the assumptions in Proposition 10 are not met. The reason is that the interior of $I_{Bx=b}$ is empty (except in the trivial case where $B = 0$ and $b = 0$). In the following proposition we present a result that shows how the quadratic bound in Proposition 10 can be improved in that case. A proof is provided in Appendix B.

Proposition 11. Assume that $f(x) = \frac{1}{2}x^THx + \xi^Tx + I_{Bx=b}(x)$ with $H \in \mathbb{S}_+^n$, $\xi \in \mathbb{R}^n$, $B \in \mathbb{R}^{p\times n}$, and $b \in \mathbb{R}^p$. Further assume that $x^THx > 0$ whenever $x \neq 0$ and $Bx = 0$, i.e. that H is positive definite on the null-space of B. Then $d : \mathbb{E}_{K^{-1}} \to \mathbb{R}$ satisfies

$$d(\mu) \leq d(\nu) + \langle \nabla d(\nu), \mu - \nu\rangle + \tfrac{1}{2}\|\mu - \nu\|_L^2 \tag{12}$$

for any $L \succeq AM_{11}A^T$ for all $\mu, \nu \in \mathbb{E}_{K^{-1}}$, where

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} = \begin{bmatrix} H & B^T \\ B & 0 \end{bmatrix}^{-1}. \tag{13}$$

Further, no matrix $L \not\succeq AM_{11}A^T$ exists such that (12) holds for all $\mu, \nu \in \mathbb{E}_{K^{-1}}$.

Note that the results in Propositions 9, 10, and 11 are independent on which space the function d is defined, i.e., they are independent of K. This holds since the shape of d is the same when defined on any $\mathbb{E}_{K^{-1}}$. The results merely relate the shape of d to $AH^{-1}A^T$. Since $AH^{-1}A^T$ might be rank deficient, this matrix cannot define a norm. However, by selecting a positive definite metric matrix K^{-1} that satisfies $K^{-1} \succeq AH^{-1}A^T$, and by linking the results in Propositions 9, 10, and 11 to the smoothness definition in Definition 3, we can summarize Propositions 9, 10, and 11 as follows, assuming that Assumption 6 holds:

- The function $d \in \Gamma_0(\mathbb{E}_{K^{-1}})$ is 1-smooth (w.r.t. $\mathbb{E}_{K^{-1}}$) for any positive definite $K^{-1} \succeq AH^{-1}A^T$.
- If f is quadratic with positive definite Hessian H on some full-dimensional set, then $d \in \Gamma_0(\mathbb{E}_{K^{-1}})$ is 1-smooth w.r.t. $\mathbb{E}_{K^{-1}}$ if and only if $K^{-1} \succeq AH^{-1}A^T$ (and K^{-1} is positive definite).
- If f is quadratic with Hessian H (that is positive definite on the nullspace of B) plus indicator function for an affine subspace $Bx = b$. Then $d \in \Gamma_0(\mathbb{E}_{K^{-1}})$ is 1-smooth (w.r.t. $\mathbb{E}_{K^{-1}}$) if and only if $K^{-1} \succeq AM_{11}A^T$ (and K^{-1} is positive definite).

In this section, we have shown how to compute tight quadratic majorizers to d . In the following section, we will show how to use this knowledge to find good metrics on which to perform dual forward-backward splitting.

5. FAST DUAL FORWARD-BACKWARD SPLITTING

We apply fast dual forward-backward splitting to solve the dual problem (5). The dual problem is defined on $\mathbb{E}_{K^{-1}}$. In this section, we introduce the notation $L = K^{-1}$, i.e., the dual problem is defined on $\mathbb{E}_L$. The algorithm is

$$\begin{aligned} \nu^k &= \mu^k + \alpha_k(\mu^k - \mu^{k+1}) \\ \mu^{k+1} &= \operatorname{prox}_{g^*}(\nu^k - L^{-1}\nabla d(\nu^k)) \end{aligned}$$

where $\alpha_k = \theta_k(\theta_{k-1}^{-1} - 1)$ and the θ_k-sequence satisfies (see Tseng (2008))

$$\frac{1}{\theta_k^2} = \frac{1 - \theta_{k+1}}{\theta_{k+1}^2} \tag{14}$$

and the prox (backward) step is defined as

$$\operatorname{prox}_{g^*}(z) := \operatorname*{argmin}_{\mu} \left\{ g^*(\mu) + \tfrac{1}{2}\|\mu - z\|_L^2 \right\}.$$

The μ^{k+1}-update can equivalently be written as (expand the square, remove constant terms and add $d(\nu^k)$ which does not affect the minimizer):

$$\operatorname*{argmin}_{\mu} \left\{ d(\nu^k) + \langle \nabla d(\nu^k), \mu - \nu^k\rangle + \tfrac{1}{2}\|\mu - \nu^k\|_L^2 + g^*(\mu) \right\}.$$

We see that d is approximated by the r.h.s. of the smoothness definition in (3), i.e., by

$$d(\nu^k) + \langle \nabla d(\nu^k), \mu - \nu^k\rangle + \tfrac{1}{2}\|\mu - \nu^k\|_L^2. \tag{15}$$

Indeed, the algorithm converges if d is 1-smooth w.r.t. the L that is used in the algorithm, see Zuo and Lin (2011), i.e., if the approximation (15) used in the algorithm is an upper bound to d. A bound on the convergence rate is (see Zuo and Lin (2011))

$$D(\mu^k) - D(\mu^\star) \leq \frac{2\|\mu^\star - \mu^0\|_L^2}{(k+1)^2} \tag{16}$$

where $D = d + g^*$ and $\mu^\star$ is a solution to (5). Also, convergence rates for the primal iterates as well as the primal infeasibility can be obtained, see Patrinos and Bemporad (2014); Nedelcu et al. (2014); Necoara and Nedelcu (2014).

Remark 12. The constant in the rate bound in (16) can be compared to the constant in the rate bound for standard fast dual forward-backward splitting used, e.g., in Richter et al. (2013), which is

$$D(\mu^k) - D(\mu^\star) \leq \frac{2\beta\|\mu^\star - \mu^0\|_2^2}{(k+1)^2}$$

where β is a Lipschitz constant for d. From the previous section (and Richter et al. (2013)), we know that $\beta = \|AH^{-1}A^T\|_2$ (which is actually tight). We compare this to (16) where we can choose $L = AH^{-1}A^T$, due to the analysis in the previous section. Therefore, the constant in the convergence rate bound provided in (16) is as good

or better than the one in Richter et al. (2013) due to Cauchy-Schwarz inequality. For ill-conditioned problems, the difference in constant can be significant, depending on the initial condition μ^0.

Remark 13. From Giselsson and Boyd (2014) it follows that forward-backward splitting algorithm applied on $\mathbb{E}_{K^{-1}}$ is equivalent to applying forward-backward splitting on the Euclidean space $\mathbb{R}^m$ to the dual of the preconditioned problem

$$\begin{array}{ll} \text{minimize} & f(x) + g(y) \\ \text{subject to} & DAx = Dy \end{array}$$

where $K = D^T D$. We can also define the dual problem on the finite-dimensional vector space with inner product $\langle x, y\rangle = x^T K y$ and with the induced norm $\|x\| = \sqrt{x^T K x}$. When applying the forward-backward splitting algorithm on this Hilbert space, we get another equivalent algorithm (note that the metric K is inverted compared to $\mathbb{E}_{K^{-1}}$). These equivalent approaches are, however, not further discussed here due to space limitations and clarity of exposition.

The important question how to choose $L = K^{-1}$ remains. As mentioned, the smooth part of the dual problem, d, is approximated by a quadratic function that majorizes it, see (15) and the r.h.s. of the smoothness definition (3). The tighter this quadratic majorizer is, the more accurate the function approximation used in the algorithm is, and consequently the faster the convergence of the algorithm is likely to be. For example, if the majorizer is tight (i.e., (3) holds with equality for all x, y), the algorithm converges in one iteration. This can be seen from the μ^{k+1}-update

$$\begin{aligned} &\operatorname*{argmin}_{\mu} \left\{ d(\nu^k) + \langle \nabla d(\nu^k), \mu - \nu^k \rangle + \tfrac{1}{2}\|\mu - \nu^k\|_L^2 + g^*(\mu) \right\} \\ &= \operatorname*{argmin}_{\mu} \left\{ d(\mu) + g^*(\mu) \right\} \end{aligned}$$

which in that case reduces to solving the original problem (5). Proposition 9 suggests that $\mathbb{E}_L$ with $L = AH^{-1}A^T$ would be a good space on which to apply the algorithm, since the r.h.s. of (3) would (in many cases) be the best quadratic majorizer of d, see Proposition 10. In most cases, however, it is not advisable to let $L = AH^{-1}A^T$ (which is typically sparse or full) since the prox operation of g^* could become computationally too expensive. For instance, if g^* is separable, using a non-diagonal L would typically increase the computational burden in each iteration more than what is gained by the reduced number of iterations. This discussion suggests that to get good performance, we should choose $L \approx AH^{-1}A^T$ where the structure of L should be such that the computational complexity of evaluating the prox operator is kept low. We also need $L \succeq AH^{-1}A^T$ to guarantee convergence of the algorithm. By letting an invertible matrix E satisfy $L = (E^T E)^{-1}$, these objectives can be formulated as choosing E such that $I \approx EAH^{-1}A^T E^T$ and $I \succeq EAH^{-1}A^T E^T$. A natural choice is then to minimize $\kappa(EAH^{-1}A^T E^T)$ subject to $I \succeq EAH^{-1}A^T E^T$, where κ denotes the condition number. However, if A does not have full row rank, or if the objective is to satisfy $L \approx AM_{11}A^T$ (when the assumptions in Proposition 11 hold), the matrix APA^T (with $P = H^{-1}$ or $P = M_{11}$) does not have a well defined condition number. In such cases, the ratio between the largest and smallest non-zero eigenvalues of $EAPA^T E^T$ could be minimized instead. This is reasonable, since the zero eigenvalues of APA^T cannot be affected or made positive by pre- and post-multiplying APA^T with E and E^T respectively. Exact methods to achieve this as well as heuristic approximations are discussed in the following section.

6. COMPUTING THE METRIC MATRIX

In this section, we show how to solve

$$\text{minimize } \frac{\lambda_1(EQE^T)}{\lambda_r(EQE^T)} \tag{17}$$

where λ_1 denotes the largest (non-zero) eigenvalue, λ_r denotes the smallest non-zero eigenvalue, and $Q \in \mathbb{S}_+^n$ is $Q = AH^{-1}A^T$ or $AM_{11}A^T$. We restrict the matrix E to be full, block-diagonal, or diagonal, since this implies that also $L = (E^T E)^{-1}$ is full, block-diagonal, or diagonal respectively. We denote by $\mathcal{E}$ any of these structural constraints. Besides showing how to solve (17) exactly, we also present heuristic methods to (hopefully) reduce the (pseudo) condition number in (17).

6.1 Exact condition number minimization

In this section, we show how to solve (17) exactly. We consider two different cases; Q positive definite, and Q positive semi-definite. In Boyd et al. (1994), it has been shown that minimizing the ratio between the largest and smallest eigenvalues of $(FRE^T)^T(FRE^T)$ by selecting E and F can be written as a quasi convex optimization problem. Here, as we will see, we are interested in the case where either $E = I$ or $F = I$. For these cases, the optimal matrices can be found by convex optimization. Before we show how to solve (17), we state the following well known lemma that follows directly from the definition of singular values.

Lemma 14. For any matrix $\Phi \in \mathbb{R}^{m\times n}$, the non-zero eigenvalues of $\Phi^T\Phi$ equals the non-zero eigenvalues of $\Phi\Phi^T$.

The positive definite case. Here, we assume that Q is positive definite, which occurs, for instance, if $Q = AH^{-1}A^T$, where $H \in \mathbb{S}_{++}^n$ and $A \in \mathbb{R}^{m\times n}$ has full row rank.

Proposition 15. Assume that $Q \in \mathbb{S}_{++}^m$. Then a matrix $E \in \mathcal{E}$ that minimizes the ratio (17) can be computed by solving the convex semi-definite program

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & tQ \succeq L \\ & Q \preceq L \\ & L \in \mathcal{E} \end{array} \tag{18}$$

where $L = (E^T E)^{-1}$. Further, $L \succeq Q$.

Proof. Since Q has full rank, (17) is the condition number. Thus, according to (Boyd et al., 1994, Section 3.1), (18) can be solved in order to minimize (17). Since the cost and constraints in (18) are all convex, this is a convex optimization problem. Further, the second constraint implies that $L \succeq Q$.

The condition number minimization problem (17) is also investigated in Lu and Pong (2011), where they search for E directly using a convex relaxation of the nonconvex

constraint $EQE^T \succeq \frac{1}{t}I$. It is shown that the convex relaxation is tight if E is diagonal. Therefore the approach in Lu and Pong (2011) is slightly more restrictive than our setting since we allow also for block-diagonal structures.

The positive semi-definite case. Here, we assume that Q is positive semi-definite. This situation occurs, e.g., if $Q = AH^{-1}A^T$ and $A \in \mathbb{R}^{m\times n}$ with $m > n$, or if $Q = AM_{11}A^T$.

Proposition 16. Assume that $Q \in \mathbb{S}^m_+$ is factorized as $Q = R^TR$, where $R \in \mathbb{R}^{q\times n}$ has rank q. Then a matrix $E \in \mathcal{E}$ that minimizes the ratio (17) can be computed by solving the convex semi-definite program

$$\begin{array}{ll} \text{minimize} & -t \\ \text{subject to} & RMR^T \preceq I \\ & RMR^T \succeq tI \\ & M \in \mathcal{E} \end{array} \tag{19}$$

where $M = (E^TE)$. Further $L = M^{-1} = (E^TE)^{-1} \succeq Q$.

Proof. Since RMR^T has full rank, we get from Lemma 14 and equalities $M = E^TE$ and $Q = R^TR$ that minimizing the condition number of RMR^T is equivalent to minimizing the ratio between the largest and smallest non-zero eigenvalues of EQE^T, i.e. equivalent to solving (17). From (Boyd et al., 1994, Section 3.1), we get that (19) minimizes the condition number of RMR^T, i.e., it minimizes (17). Since the cost and constraints in (19) are all convex, this is a convex optimization problem. Further, the first inequality in (19) implies through Lemma 14 that $EQE^T \preceq I$, which is equivalent to that $L = (E^TE)^{-1} \succeq Q$. This concludes the proof.

6.2 Heuristic 1 – Trace minimization

We also propose to use a trace minimization heuristic to reduce the (pseudo) condition number of EQE^T. Let $L = (E^TE)^{-1}$ to get

$$\begin{array}{ll} \text{minimize} & \text{trace } L \\ \text{subject to} & Q \preceq L \\ & L \in \mathcal{E} \end{array}$$

Also this is a semi-definite program and therefore restricted to small-scale problems.

6.3 Heuristic 2 – Equilibration

In equilibration, given a matrix $T \in \mathbb{R}^{m\times n}$, the objective is to find positive and diagonal matrices $E \in \mathbb{R}^{m\times m}$ and $F \in \mathbb{R}^{n\times n}$ such that all rows and columns of the scaled matrix ETF have the same length in some norm. This is a heuristic to reduce the condition number of the scaled matrix ETF compared to the original matrix T, see Bradley (2010) for an overview of equilibration and further references. There are no guarantees that the condition number is reduced, but in practice, this is most often the case. In our setting, we will use this heuristic to improve the conditioning of EQE^T which is the objective stated in (17). To this end, we consider symmetric equilibration where the pre- and post-multiplied matrices are the same, i.e., where $E = F$. Below, we present different methods to achieve symmetric equilibration of symmetric matrices in the 1-norm, 2-norm and ∞-norm. None of these methods guarantee that $I \succeq EQE^T$, which is required to get convergence of the fast dual forward-backward splitting method. However, this is achieved by appropriately scaling E using a norm computation.

Equilibration in 1-norm and 2-norm. For a symmetric matrix, the ith row and column are the same, hence also their norms. Thus, in symmetric equilibration of symmetric matrices, we need only equilibrate either the rows or the columns. The 1-norm of row i of EQE^T is given by

$$\left\|[EQE^T]_{i,\cdot}\right\|_1 = \sum_{j=1}^m |E_{ii}Q_{ij}E_{jj}| = E_{ii}\sum_{j=1}^m |Q_{ij}|\,E_{jj}$$

since E is diagonal with $E_{ii} > 0$. Similarly, the squared 2-norm is given by

$$\left\|[EQE^T]_{i,\cdot}\right\|_2^2 = \sum_{j=1}^m (E_{ii}Q_{ij}E_{jj})^2 = E_{ii}^2\sum_{j=1}^m Q_{ij}^2E_{jj}^2$$

Thus, by introducing the matrices $T_1 = |Q|$ (where $|\cdot|$ denotes element-wise absolute value) and $T_2 = (Q)^{(2)}$ (where $(\cdot)^{(2)}$ denotes element-wise square), and by letting $E = \mathrm{diag}(e)$, symmetric equilibration can be stated as finding E (and e) such that

$$ET_1e = \mathbf{1} \tag{20}$$

in the 1-norm case and

$$E^{(2)}T_2e^{(2)} = \mathbf{1} \tag{21}$$

in the 2-norm case. We treat these cases simultaneously by introducing $\widetilde{E} = \mathrm{diag}(\widetilde{e})$ and T that satisfies $\widetilde{E} = E$ and $T = T_1$ in the 1-norm case, and $\widetilde{E} = E^{(2)}$ and $T = T_2$ in the 2-norm case. The conditions (20) and (21) can then be written as

$$0 = T\widetilde{e} - \widetilde{E}^{-1}\mathbf{1}.$$

This is indeed the gradient of the function

$$\phi(\widetilde{e}) = \tfrac{1}{2}\widetilde{e}^TT\widetilde{e} - \sum_{i=1}^n \ln(\widetilde{e}_i). \tag{22}$$

Since $\widetilde{e}^TT\widetilde{e} \geq \sum_i(T_{ii}\widetilde{e}_{ii}^2) \geq (\min_i T_{ii})\|\widetilde{e}\|_2^2$ for all $\widetilde{e} \in \mathrm{dom}\phi$, i.e. for all $\widetilde{e} > 0$, and since $-\ln$ is convex, ϕ is convex on its domain. If in addition $\min_i T_{ii} > 0$ (which is the case we are interested in), then ϕ is strongly convex. Since $\phi(\widetilde{e}) < \infty$ for all $\widetilde{e} \in \mathrm{int}(\mathrm{dom}\phi)$ and since $\phi(\widetilde{e}) \to \infty$ as $\widetilde{e}$ approaches the boundary of the domain, we conclude that ϕ has a unique minimizer $\widetilde{e}^\star \in \mathrm{int}(\mathrm{dom}\phi)$. This unique minimizer can be found in various ways.

One approach is to perform element-wise optimization and cycle through the elements until convergence. Optimality conditions for optimizing (22) w.r.t. $\widetilde{e}_i$, while the other $\widetilde{e}_j$ are fixed, are given by

$$\begin{aligned} & 0 = T_{ii}\widetilde{e}_i + \sum_{j\neq i} T_{ij}\widetilde{e}_j + 1/\widetilde{e}_i \\ \Leftrightarrow \quad & 0 = T_{ii}\widetilde{e}_i^2 + (\sum_{j\neq i} T_{ij}\widetilde{e}_j)\widetilde{e}_i + 1 \end{aligned}$$

Each such element-wise optimization is very cheap since it requires the solution of a second-order equation and a vector-vector multiplication of size $m-1$.

A classic method to perform 1-norm and 2-norm equilibration is the (symmetric) Sinkhorn-Knopp algorithm, Sinkhorn and Knopp (1967), which was originally developed to generate doubly stochastic matrices from positive

matrices. The symmetric Sinkhorn-Knopp algorithm is given by the iteration

$$\widetilde{e}^{k+1} = (T\widetilde{e}^{k})^{-1}$$

where $(\cdot)^{-1}$ denotes element-wise reciprocal. This is known to converge, see Sinkhorn and Knopp (1967), to an alternating sequence that satisfies $\widetilde{e}^{k} = \beta\widetilde{e}^{k+1}$ for some $\beta \in \mathbb{R}_{++}$ under the conditions that T is positive, symmetric, and fully indecomposable. Full indecomposability means that no sub-matrix with only zeros of size $l \times p$ exists where $l + p \geq m$, where m is the dimension of T. This excludes, e.g., block-diagonal matrices where each block instead have to be equilibrated separately.

Also other equilibration methods exist, see Ruiz (2001); Bradley (2010); Knight et al. (2011). Common for all these methods is that usually two to five passes over the data are sufficient to obtain a close to equilibrated matrix. This implies that they are computationally very cheap, and can be used both for offline and online metric selection.

Equilibration in ∞-norm. In ∞-norm equilibration of general symmetric matrices, the magnitude of the largest element in each row (or column) is set to 1. For positive semi-definite matrices $S \in \mathbb{S}^n_+$, we have that $S_{ii} \geq 0$ and $\max_i S_{ii} \geq \max_{i \neq j} |S_{ij}|$, see (Horn and Johnson, 1990, p. 398). Thus, for positive semi-definite matrices with $S_{ii} > 0$ for all i, having $S_{ii} = 1$ for all i gives equilibration in the ∞-norm. For $S = EQE^T$ with Q positive semi-definite with positive diagonal, this scaling, which is also referred to Jacobi scaling, is obtained by letting $E_{ii} = 1/\sqrt{Q_{ii}}$. This is less computationally expensive than equilibration in the 1-norm or 2-norm as discussed above.

7. APPLICATIONS

In this section, we show how to apply the results in this paper on two quadratic programming formulations. We consider one formulation with no specific structure that we solve in two different ways, and one that has a separable structure that allows for distributed implementation. We state the algorithms on $\mathbb{E}_{K^{-1}}$ and let $L = K^{-1}$ to simplify notation.

7.1 Two QP splittings

Here, we consider the following quadratic program

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T H x + \xi^T x \\ \text{subject to} & Bx = b \\ & \underline{d} \leq Cx \leq \bar{d} \end{array}$$

where $H \in \mathbb{S}^n_{++}$, $\xi \in \mathbb{R}^n$, $B \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $C \in \mathbb{R}^{p \times n}$, and $\underline{d}, \bar{d} \in \mathbb{R}^p$. We form two different dual problems to this problem by employing two different splittings that has previously been used in the literature. We present the resulting fast dual forward-backward splitting method and discuss how to choose the space on which to apply the algorithm.

QP splitting 1. In the first splitting, which has been used in Patrinos and Bemporad (2014) in the context of fast dual forward-backward splitting and in O'Donoghue et al. (2013) in the context of ADMM, we set f and g in (4) to

$$\begin{aligned} f(x) &= \tfrac{1}{2}x^T H x + \xi^T x + I_{Bx=b}(x) \\ g(y) &= I_{\underline{d} \leq y \leq \bar{d}}(y) \end{aligned}$$

which gives equality constraint $Cx = y$. (Here, it is enough to assume that H is positive definite on the null-space of B.) We form the dual as in (5) and restrict $L = K^{-1}$ in the algorithm to be diagonal. After some simplification, the resulting fast dual forward-backward splitting algorithm on $\mathbb{E}_L = \mathbb{E}_{K^{-1}}$ becomes

$$\nu^k = \mu^k + \alpha_k(\mu^k - \mu^{k-1}) \tag{23}$$

$$x^k = \operatorname*{argmin}_x \left\{ \tfrac{1}{2}x^T H x + \xi^T x + I_{Bx=b}(x) + \nu^T C x \right\} \tag{24}$$

$$\mu^k = \min\Big(\nu^k + L^{-1}(Cx^k - \underline{d}), \max\left(\nu^k + L^{-1}(Cx^k - \bar{d}), 0\right)\Big) \tag{25}$$

where $\alpha_k = \theta_k(\theta_{k-1}^{-1} - 1)$ and θ_k satisfies (14). The restriction that L is diagonal implies that the prox-operation becomes a min-max operation only, hence very cheap. The matrix L should be computed in accordance with the suggestions in Section 6. Specifically, according to Proposition 11, we need $L \succeq CM_{11}C^T$, where M_{11} is defined in (13). Obviously also $L \succeq CH^{-1}C^T$ holds since $M_{11} \preceq H^{-1}$.

Equation (24) can be efficiently implemented since it is an equality constrained quadratic problem. It can be solved by forming and storing M_{11} and M_{12} in (13), and reuse these in all iterations. Another option, that might be beneficial if $\left[\begin{smallmatrix} H & B^T \\ B & 0 \end{smallmatrix}\right]$ (from (13)) is sparse, is to compute and store a sparse LDL factorization of $\left[\begin{smallmatrix} H & B^T \\ B & 0 \end{smallmatrix}\right]$ and perform forward and backward substitution on the factorization in each subsequent iteration.

QP splitting 2. The second splitting, which has been used in Richter et al. (2013), is obtained by letting f and g in (4) be

$$\begin{aligned} f(x) &= \tfrac{1}{2}x^T H x + \xi^T x + I_{\underline{d} \leq Cx \leq \bar{d}}(x) \\ g(y) &= I_{y=b}(y) \end{aligned}$$

which gives equality constraint $Bx = y$. Without structural restrictions on $L = K^{-1}$, the resulting fast dual forward backward splitting method on $\mathbb{E}_L = \mathbb{E}_{K^{-1}}$ becomes after some simplification:

$$\nu^{k+1} = \mu^k + \alpha_k(\mu^k - \mu^{k-1}) \tag{26}$$

$$x^k = \operatorname*{argmin}_x \left\{ \tfrac{1}{2}x^T H x + \xi^T x + I_{\underline{d} \leq Cx \leq \bar{d}}(x) + \nu^T B x \right\} \tag{27}$$

$$\mu^k = \nu^k + L^{-1}(Bx^k - b) \tag{28}$$

Since we have no structural constraints on L, we can choose any $L \succeq BH^{-1}B^T$. If $BH^{-1}B^T$ is sparse, an efficient choice is to let $L = BH^{-1}B^T$ and compute and store a sparse Cholesky factorization of $BH^{-1}B^T$. Updating μ^k then reduces to a forward and backward solve in each subsequent iteration.

The complexity of solving (27) depends highly on the structures of H and C. If H and C are block-diagonal with sufficiently small blocks, then (27) can be solved efficiently and exactly in parallel using, e.g., the MPT toolbox, Herceg et al. (2013). In the limiting case where H and C are diagonal, solving (27) reduces to one max and one min operation for each variable. For problems

where solving (27) is computationally expensive and no exact solutions can be obtained easily or fast, we suggest to instead use QP splitting 1.

7.2 The distributed case

Here, we consider separable optimization problems of the form (which is similar to the formulations in Beck et al. (2013); Necoara and Nedelcu (2015)):

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{M} (f_i(x_i) + g_i(y_i)) \\ \text{subject to} & Ax = y \end{array} \tag{29}$$

where $f_i : \mathbb{E}_{H_i} \to \overline{\mathbb{R}}$ is 1-strongly convex (w.r.t. $\mathbb{E}_{H_i}$), $g_i \in \Gamma_0(\mathbb{E}_{K_i})$, $x = (x_1, \ldots, x_M)$, $y = (y_1, \ldots, y_M)$, and

$$A = \begin{bmatrix} A_{11} & \ldots & A_{1M} \\ \vdots & \ddots & \vdots \\ A_{M1} & \ldots & A_{MM} \end{bmatrix}.$$

We further assume that many $A_{ij} = 0$. The non-zero block entries of A are indexed by the sets

$$\begin{aligned} \mathcal{N}_i &= \{j \in \{1, \ldots, M\} \mid A_{ij} \neq 0\} \\ \mathcal{M}_j &= \{i \in \{1, \ldots, M\} \mid A_{ij} \neq 0\}. \end{aligned}$$

We introduce the notation $x_{\mathcal{N}_i} = (\ldots, x_j, \ldots)$ that stacks all x_j with $j \in \mathcal{N}_i$, and $A_{\mathcal{N}_i} = [\ldots, A_{ij}, \ldots]$ that collects all $A_{ij} \neq 0$ in block-row i. This implies that (29) can equivalently be written as

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{M} (f_i(x_i) + g_i(y_i)) \\ \text{subject to} & A_{\mathcal{N}_i} x_{\mathcal{N}_i} = y_i, \text{ for all } i \in \{1, \ldots, M\} \end{array}$$

We introduce dual variables μ_i for all equality constraints $A_{\mathcal{N}_i} x_{\mathcal{N}_i} = y_i$ and define the vectors $\mu_{\mathcal{M}_j} = (\ldots, \mu_i, \ldots)$ that stacks all μ_i with $i \in \mathcal{M}_j$. We also define the matrix $A_{\mathcal{M}_j} = [\ldots, A_{ij}^T, \ldots]^T$ that collects all $A_{ij} \neq 0$ in block-column j. This implies that the dual problem can be written as

$$\text{minimize} \quad \sum_{i=1}^{M} (d_i(\mu_{\mathcal{M}_i}) + g_i^\star(\mu_i))$$

where

$$d_i(\mu_{\mathcal{M}_i}) := f_i^\star(-A_{\mathcal{M}_i}^T \mu_{\mathcal{M}_i}).$$

In Beck et al. (2013), it is shown how to compute a matrix L that defines a space $\mathbb{E}_L$, using distributed computations only, on which

$$d(\mu) := \sum_{i=1}^{m} d_i(\mu_{\mathcal{M}_i})$$

is 1-smooth. The procedure from Beck et al. (2013) is presented below:

$$L_{\mathcal{M}_j} = \frac{\|A_{\mathcal{M}_j}\|_2^2}{\lambda_{\min}(H_j)} \tag{30}$$

$$L_i = \sum_{j \in \mathcal{N}_i} L_{\mathcal{M}_j} \tag{31}$$

$$L = \text{blkdiag}(L_1 I, \ldots, L_M I) \tag{32}$$

This metric selection procedure relies on that d_i is $\frac{\|A_{\mathcal{M}_i}\|_2^2}{\lambda_{\min}(H_i)}$-smooth w.r.t $\mathbb{E}_I$. From Proposition 9, we know that d_i is majorized by a quadratic with Hessian $A_{\mathcal{M}_i} H_i^{-1} A_{\mathcal{M}_i}^T$, i.e. that

$$\begin{aligned} d_i(\mu_{\mathcal{M}_i}) \leq d_i(\nu_{\mathcal{M}_i}) + \langle \nabla d_i(\nu_{\mathcal{M}_i}), \mu_{\mathcal{M}_i} - \nu_{\mathcal{M}_i} \rangle_2 \\ + \tfrac{1}{2} \|\mu_{\mathcal{M}_i} - \nu_{\mathcal{M}_i}\|^2_{A_{\mathcal{M}_i} H_i^{-1} A_{\mathcal{M}_i}^T}, \end{aligned}$$

or equivalently that d_i is 1-smooth w.r.t. $\mathbb{E}_{L_{\mathcal{M}_i}}$ for any $L_{\mathcal{M}_i} \succeq A_{\mathcal{M}_i} H_i^{-1} A_{\mathcal{M}_i}^T$. This is obviously a tighter characterization of d_i than the one used in Beck et al. (2013), i.e., that d_i is $\frac{\|A_{\mathcal{M}_i}\|_2^2}{\lambda_{\min}(H_i)}$-smooth w.r.t. $\mathbb{E}_I$. Using Proposition 9, the distributed metric selection procedure proposed in Beck et al. (2013) can be modified to yield a less conservative set from which L may be chosen. The modified procedure is presented below:

$$L_{\mathcal{M}_j} = \text{blkdiag}(\ldots, L_{\mathcal{M}_j, i}, \ldots) \succeq A_{\mathcal{M}_j} H_j^{-1} A_{\mathcal{M}_j}^T \tag{33}$$

$$L_i = \sum_{j \in \mathcal{N}_i} L_{\mathcal{M}_j, i} \tag{34}$$

$$L = \text{blkdiag}(L_1, \ldots, L_M) \tag{35}$$

where $L_{\mathcal{M}_j, i}$ are sub-blocks of the same dimension as μ_i. With this restriction, is it straight-forward to verify that the fast dual forward backward splitting algorithm on $\mathbb{E}_L$ can be implemented in distributed fashion with the same communication structure as the algorithm in Beck et al. (2013). Also, the proof in Beck et al. (2013) to show that d is 1-smooth w.r.t. $\mathbb{E}_L$ with L computed as in (30)-(32) is straight-forwardly generalized to show that d is 1-smooth w.r.t. $\mathbb{E}_L$ with L as computed in (33)-(35). This is needed to guarantee convergence of the distributed fast dual forward-backward splitting algorithm.

By choosing $L_{\mathcal{M}_j}$ appropriately (using some method from Section 6) an L computed by (33)-(35) probably gives a better approximation of d (in the sense of (3) with $\|\cdot\|_L$) than if L is computed using (30)-(32). Computing L from (33)-(35) is therefore likely to improve the convergence of the algorithm compared to using (30)-(32), i.e., the method in Beck et al. (2013). This is confirmed by numerical examples in Section 8.

Remark 17. Note that $L_{\mathcal{M}_j}$ is restricted to be block-diagonal to facilitate a distributed implementation. However, we could superimpose additional internal structural constraints on each sub-block $L_{\mathcal{M}_j, i}$ that may differ from one block to the next. For instance, each sub-block $L_{\mathcal{M}_j, i}$ could be restricted to be diagonal, sparse, or full.

Remark 18. Applying the dual fast forward backward splitting on $\mathbb{E}_L$ with L computed as in (33)-(35), enables for a distributed implementation (since the resulting L is block-diagonal). If $g = I_{y=b}(y)$, the problem has similar structure as QP-splitting 2 in Section 7.1. Then, as in QP-slitting 2, we can choose $L = AH^{-1}A^T$ since no prox-operation is needed for the dual variable update. This choice of L gives a parallel algorithm where the primal variables are updated in parallel, and the dual variable updates require centralized computations. The centralized computations reduce to one forward and one backward substitution after an initial Cholesky factorization of $AH^{-1}A^T$.

8. NUMERICAL EXAMPLES

In this section, we present numerical evaluations of the proposed methods. We apply the methods on a (small-

Table 1. Comparison to other first-order methods, all implemented in MATLAB. FDFBS refers to fast dual forward backward splitting, ADMM refers to alternating direction method of multipliers, QPi refers to QP-splitting i (for $i = 1, 2$).

		exec time (ms)		nbr iters	
Algorithm/splitting(/reference)	Parameters	avg.	max	avg.	max
FDFBS/QP1	L diag fr. (19) with $Q = CM_{11}C^T$	1.4	7.1	23.5	128
FDFBS/QP1	L diag fr. (18) with $Q = CH^{-1}C^T$	1.2	5.8	20.0	105
FDFBS/QP1/Patrinos and Bemporad (2014)	$L = \|CM_{11}C^T\|_2 I$	98.5	673.0	1835.9	12686
FDFBS/QP1/Patrinos and Bemporad (2014)	$L = \|CH^{-1}C^T\|_2 I$	98.9	679.4	1850.1	12783
FDFBS/QP2	$L = BH^{-1}B^T$	2.3	12.1	21.7	102
FDFBS/QP2/Richter et al. (2013)	$L = \|BH^{-1}B^T\|_2 I$	4713.9	28411	50845	308210
ADMM/QP1/O'Donoghue et al. (2013)	$\rho = 0.3$	193.9	920.6	3129.5	15037
ADMM/QP1/O'Donoghue et al. (2013)	$\rho = 3$	29.7	142.2	457.3	2179
ADMM/QP1/O'Donoghue et al. (2013)	$\rho = 30$	35.1	264.4	556.7	4194

scale) aircraft control problem and on large-scale separable randomly generated problems that are solved in distributed fashion.

8.1 Aircraft control

Here, we apply QP-splitting 1 and QP-splitting 2 from Section 7.1 to the AFTI-16 aircraft model in Kapasouris et al. (1990); Bemporad et al. (1997). As in Bemporad et al. (1997), the continuous time model from Kapasouris et al. (1990) is sampled using zero-order hold every 0.05 s. The system has four states $x = (x_1, x_2, x_3, x_4)$, two outputs $y = (y_1, y_2)$, two inputs $u = (u_1, u_2)$, and obeys the following dynamics

$$x^+ = \begin{bmatrix} 0.999 & -3.008 & -0.113 & -1.608 \\ -0.000 & 0.986 & 0.048 & 0.000 \\ 0.000 & 2.083 & 1.009 & -0.000 \\ 0.000 & 0.053 & 0.050 & 1.000 \end{bmatrix} x + \begin{bmatrix} -0.080 & -0.635 \\ -0.029 & -0.014 \\ -0.868 & -0.092 \\ -0.022 & -0.002 \end{bmatrix} u,$$
$$y = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x$$

where x^+ denotes the state in the next time step. The system is unstable, the magnitude of the largest eigenvalue of the dynamics matrix is 1.313. The outputs are the attack and pitch angles, while the inputs are the elevator and flaperon angles. The inputs are physically constrained to satisfy $|u_i| \leq 25^\circ$, $i = 1, 2$. The outputs are soft constrained to satisfy $-s_1 - 0.5 \leq y_1 \leq 0.5 + s_2$ and $-s_3 - 100 \leq y_2 \leq 100 + s_4$ respectively, where $s = (s_1, s_2, s_3, s_4) \geq 0$ are slack variables. The cost in each time step is

$$\ell(x, u, s) = \frac{1}{2}\big((x - x_r)^T Q(x - x_r) + u^T R u + s^T S s\big)$$

where x_r is a reference, $Q = \mathrm{diag}(10^{-4}, 10^2, 10^{-3}, 10^2)$, $R = 10^{-2}I$, and $S = 10^6 I$. This gives a condition number of 10^{10} of the full cost matrix. Further, the terminal cost is Q, and the control and prediction horizon is $N = 10$. The numerical data in Tables 1 and 2 are obtained by following a reference trajectory on the output. The objective is to change the pitch angle from 0° to 10° and then back to 0° while the angle of attack satisfies the output constraints $-0.5^\circ \leq y_1 \leq 0.5^\circ$. The constraints on the angle of attack limits the rate on how fast the pitch angle can be changed. The full optimization problem can be written on the form

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} z^T H z + r_t^T z \\ \text{subject to} & \bar{B} z = b x_t \\ & \underline{d} \leq C z \leq \bar{d} \end{array} \tag{36}$$

where x_t and r_t change from one sampling instant to the next.

In Table 1 we compare the performance of QP-splitting 1 and QP-splitting 2 from Section 7.1, when solving (36). We compare the performance of these when applied in the Euclidean setting $\mathbb{E}_I$ and when applied on a space $\mathbb{E}_L$, where L is computed as suggested in Section 6. Since QP-splitting 1 in the Euclidean space is exactly the algorithm proposed in Patrinos and Bemporad (2014), and QP-splitting 2 in the Euclidean space is exactly the algorithm proposed in Richter et al. (2013), the comparison in Table 1 is a comparison to other methods proposed for embedded model predictive control in the literature. We also compare the performance of these methods with the performance of the ADMM-based method for embedded model predictive control proposed in O'Donoghue et al. (2013). Since the method in O'Donoghue et al. (2013) is based on ADMM, the ρ-parameter much be chosen. In Table 1 we provide simulation results for three different values of ρ, the best performing ρ that we found ($\rho = 3$) and one larger and one smaller value of ρ. All algorithms are implemented in MATLAB and the numerical results in Table 1 are obtained by running the simulations on a Linux machine using a single core running at 2.9 GHz. To create an easily transferable and fair termination criterion, the solution to each optimization problem $z^\star$ is computed to high accuracy using an interior point solver. The optimality condition is then set to $\|z^k - z^\star\|_2 / \|z^\star\|_2 \leq 0.005$, where z^k is the primal iterate in the algorithm at iteration k. This stopping criterion implies that a relative accuracy of 0.5% of the primal solution is required.

Table 1 indicates that applying the fast forward backward splitting method for this example using QP-splitting 1 on $\mathbb{E}_L$, where L is diagonal and computed by minimizing the (pseudo) condition number of CH^1C^T or $CM_{11}C^T$, outperforms with one to two orders of magnitude QP-splitting 1 applied in the Euclidean space, i.e., the algorithm in Patrinos and Bemporad (2014). From Table 1 we also read that QP-splitting 2 applied on $\mathbb{E}_L$ with $L = BH^{-1}B^T$ for this example outperforms the algorithm in Richter et al. (2013), i.e., QP-splitting 2 on the Euclidean space, with more than three orders of magnitude. We also see that the fast dual forward-backward splittings methods applied on appropriate spaces converge considerably faster than the method in O'Donoghue et al. (2013).

In Table 2, we compare different solvers implemented in C. For QP-splitting 1 we use QPgen Giselsson (2014a) which implements this methods. For QP-splitting 2, we generate C code that take the reference trajectory and

Table 2. Comparison to other solvers, all implemented in C. We report average and worst case execution times, code size for methods that generate problem specific code, and what type of algorithm that is used.

		exec time (ms)			
Alg.(/split.)	Comments	avg.	max	code size	algorithm type
FDFBS/QP1 (QPgen)	L diag fr. (19) with $Q = CH^{-1}C^T$	0.083	0.212	36 kB	First order
FDFBS/QP2	$L = BH^{-1}B^T$	0.079	0.232	54 kB	First order
FORCES		0.347	0.592	109 kB	Interior-point
CVXGEN		0.639	0.760	404 kB	Interior-point
MPT toolbox	$N = 3$ and no tracking	0.22	0.31	9.8 MB	Explicit
qpOASES	warm-starting version	0.189	5.8	-	Online active set
qpOASES	cold-starting version	4.7	6.0	-	Active set
DuQuad	alg: inexact fast dual grad. method	27.2s	56.5s	-	First order
FiOrdOs	alg: Chambolle and Pock (2011)	38.4	58.2	27 kB	First order
MOSEK		4.6	8.1	-	Interior-point

the initial state as inputs. Compared to the corresponding MATLAB implementations in Table 1, the generated C code is more than 20 times faster. Also, the two different QP-splittings have similar performance. These implementations are compared to other toolboxes written in C for MPC problems, namely FORCES, CVXGEN, qpOASES, FiOrdOs, DuQuad, the MPC toolbox. We also compare to the commercial software MOSEK. FORCES Domahidi et al. (2012) and CVXGEN Mattingley and Boyd (2012) are based on interior point methods. We see that our methods are three to five times faster for this problem. DuQuad Necoara and Patrascu (2015) and FiOrdOs Ullmann and Richter (2012) are based on first order methods with inexact (DuQuad) or exact (FiOrdOs) inner minimizations. In DuQuad, the fast gradient method is chosen, and in FiOrdOs, the primal-dual method in Chambolle and Pock (2011) is chosen. We see that our methods perform quite much better than these methods. This is largely due to the developed preconditioning techniques. Actually, DuQuad reached the maximum number of iterations (set to 10^7 inner and 10^6 outer iterations) in more than 50% of the problem instances. For the remaining instances (that was solved to sufficient accuracy) the average execution time was 0.15s and the worst case execution time was 6.56s. We also compare to qpOASES, which has similar performance as our methods for this problem in the warm-starting case. The problem is too big for the MPT toolbox. However, we have compared to the MPT toolbox when the horizon is shortened to three (instead of ten), and when no reference tracking is used. This gives 30 instead of 100 decision variables and 4 instead of 64 parameters. Even in this reduced setting, the execution time is not better than for our methods and the code size is much larger. We note that this is close to the upper size limit of what the multi-parametric approach can handle. Finally, we compare to the commercial solver MOSEK, which is more than one order of magnitude slower than our methods.

8.2 Distributed examples

Here, we apply the fast dual forward-backward splitting method to randomly generated dynamical systems with a sparse dynamic interaction. The dynamic interaction structure is decided using the method in (Kraning et al., 2013 §6.1) and the number of subsystems are 500, 2000, and 8000 respectively. The resulting average degree of the generated interconnection structures are 2.27, 2.23, and 2.23 respectively. The number of states in each subsystem is randomly chosen from the interval $\{10, 11, \ldots, 20\}$, the number of inputs are three or four, and the control horizon is $N = 10$. This gives a total number of 87060, 350860, and 1405790 decision variables respectively. The entries of the dynamics and input matrices are randomly chosen from the intervals $[-0.7\ 1.3]$ and $[-1\ 1]$ respectively. Then the dynamics matrix is re-scaled to get a spectral radius of 1.15. The states and inputs are upper and lower bounded by random bounds generated from the intervals $[0.4\ 1]$ and $[-1\ -0.4]$ respectively. The state and input cost matrices are diagonal and each diagonal entry is randomly chosen from the interval $[1\ 10^6]$.

We evaluate the distributed fast dual forward backward splitting when applied on the Euclidean space, when applied on $\mathbb{E}_L$ where L is computed as in (30)-(32) (this is the method as proposed in Beck et al. (2013)), and when applied on $\mathbb{E}_L$ where L is computed by the procedure presented in this paper, (33)-(35). These methods are compared to the parallel method discussed in Remark 18 and to the dual Newton conjugate gradient (CG) method proposed in Kozma et al. (2014). The dual Newton CG method presented in Kozma et al. (2014) solves the dual problem using a Newton method. The search direction is computed by solving the resulting linear equations to some accuracy using distributed conjugate gradient iterations. In each of these iterations, one local and two global communications are performed. The Newton step-size is decided by a distributed line search procedure that requires two global communications for each function value comparison. In the algorithm, the accuracy of the solution to the linear system solved by the conjugate gradient method must be specified. There is a trade-off between the number of iterations in the CG-algorithm and the quality of the resulting search direction. If the accuracy requirement is too low, we get close to a gradient direction, which results in an expensive method that takes approximately gradient steps. On the other hand, if the accuracy requirement is too high, too many CG-iterations are performed in each iteration which gives a high communication load.

The evaluation in Table 3 is obtained by generating 200 feasible random initial conditions from the state constraint set for each of the systems. The corresponding optimal control problems are solved using the different algorithms, each utilizing 12 parallel cores. For each problem size, we compare the performance when the L-matrix (that defines $\mathbb{E}_L$) is computed using (33)-(35), where in (33) we use dif-

Table 3. Numerical evaluation for fast dual forward-backward splitting (FDFBS) applied on different spaces $\mathbb{E}_L$, including the space selection from Beck et al. (2013) and the Euclidean space. The comparison also includes the dual Newton CG method in Kozma et al. (2014).

			# communication rounds				
			local		global		avg. exec. time
Algorithm	Parameters	# ss/vars./constr.	avg.	max	avg.	max	12 cores [mm:ss.s]
FDFBS	$L = AH^{-1}A^T$	500/87k/246k	-	-	16.2	118	2.3
FDFBS	L fr. (33)-(35), min trace	500/87k/246k	523.7	774	-	-	3.2
FDFBS	L fr. (33)-(35), $\|\cdot\|_2$-equil.	500/87k/246k	1912.5	3022	-	-	10.1
Beck et al. (2013)	L fr. (30)-(32)	500/87k/246k	4789.8	7558	-	-	25.4
FDFBS	$L = \|AH^{-1}A^T\|_2 I$	500/87k/246k	6114.7	6556	-	-	32.4
Kozma et al. (2014)	$\epsilon_i = 10^{-4}, \mu = 0.8, \sigma = 0.3$	500/87k/246k	6661.1	28868	4082.6	17694	2:06.0
FDFBS	$L = AH^{-1}A^T$	2000/351k/993k	-	-	4.5	12	7.9
FDFBS	L fr. (33)-(35), min trace	2000/351k/993k	356.8	652	-	-	15.6
FDFBS	L fr. (33)-(35), $\|\cdot\|_2$-equil.	2000/351k/993k	1138.0	1666	-	-	33.0
Beck et al. (2013)	L fr. (30)-(32)	2000/351k/993k	2530.5	3218	-	-	1:13.5
FDFBS	$L = \|AH^{-1}A^T\|_2 I$	2000/351k/993k	4474.9	4608	-	-	2:09.9
Kozma et al. (2014)	$\epsilon_i = 10^{-4}, \mu = 0.8, \sigma = 0.3$	2000/351k/993k	6464.1	20624	3961.9	12641	41:28.0
FDFBS	$L = AH^{-1}A^T$	8000/1.41M/3.98M	-	-	2.0	2	9.4
FDFBS	L (33)-(35), min trace	8000/1.41M/3.98M	340.2	426	-	-	44.6
FDFBS	L fr. (33)-(35), $\|\cdot\|_2$-equil.	8000/1.41M/3.98M	1350.1	1776	-	-	2:10.8
Beck et al. (2013)	L fr. (30)-(32)	8000/1.41M/3.98M	3050.2	3458	-	-	4:55.5
FDFBS	$L = \|AH^{-1}A^T\|_2 I$	8000/1.41M/3.98M	10583.4	10688	-	-	17:05.3

ferent methods from Section 6. We have chosen to include the cases where the local $L_{\mathcal{M}_j}$ in (33) are computed by trace-minimization and 2-norm equilibration only. Computing $L_{\mathcal{M}_j}$ by minimizing the (pseudo) condition number is not included due to computational complexity, and equilibration in 1-norm and ∞-norm is also excluded since the performance is very similar to the 2-norm equilibration case. The performance of these methods is compared to the performance of the method from Beck et al. (2013), i.e., when the L-matrix is computed using (30)-(32), and to fast dual decomposition with the optimal parameter selection given by $L = \|AH^{-1}A^T\|_2 I$. We see that the parallel algorithm has fewest communications rounds (which is two times the iteration count for these algorithms), then the algorithm with L computed using trace minimization, then L computed using equilibration, thereafter the method from Beck et al. (2013), and finally fast dual decomposition with a global step-size. This is indeed expected, since as we traverse up the list, the approximation used in the algorithm that is defined by the L-matrix matches better and better the smooth part of the dual function, d. Note that the algorithm with fewest communication rounds is a parallel algorithm, i.e., L has no block-diagonal structure. For the algorithm where the L-matrix is computed using local trace-minimization problems, the sub-blocks in L have a sparsity pattern that is not diagonal, while the remaining algorithms have diagonal L-matrices. The added flexibility in having more non-zero elements in L gives a reduced number of iterations. We also note that the dual Newton CG method in Kozma et al. (2014) performs considerably worse than the other methods on this example.

The execution times in Table 3 are pure execution times for the DMPC scheme, i.e., without offline pre-processing steps such as metric selection and factorization of matrices. To solve a semi-definite program, or to factorize a very large matrix to find the metric L is usually only computationally beneficial if the resulting L can be used to solve many optimization problems, as is the case in DMPC. Thus the first two rows in Table 3 of every problem size are merely for DMPC applications, or other applications with the same characteristic. The computational complexity for equilibration is only slightly higher than for the method in Beck et al. (2013). Both these methods are considerably less computationally demanding than computing $\|AH^{-1}A^T\|_2$ which is required to compute a global step-size. This implies that these methods are very useful when selecting metric also in general distributed optimization.

9. CONCLUSIONS

We have proposed several methods, with different computational complexity, to compute spaces on which to perform fast dual forward-backward splitting, when the primal optimization problem is strongly convex. We have evaluated these methods by applying them to an aircraft control problem and to distributed optimization problems. For the most ill-conditioned problem, the numerical evaluations show that it is possible to reduce the computational effort by up to three orders of magnitude compared to applying the algorithms on the Euclidean space.

REFERENCES

Mosek ApS. Optimization software, 2013. `http://www.mosek.com/`.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, October 2009.

A. Beck, A. Nedic, A. Ozdaglar, and M. Teboulle. Optimal distributed gradient methods for network resource allocation problems. Submitted for publication, 2013.

A. Bemporad, A. Casavola, and E. Mosca. Nonlinear control of constrained linear systems via predictive reference management. *IEEE Transactions on Automatic Control*, 42(3):340–349, 1997.

A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, January 2002.

J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, 2004.

S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, June 1994.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

A. Bradley. *Algorithms for the Equilibration of Matrices and Their Application to Limited-Memory Quasi-Newton Methods*. PhD thesis, Stanford University, 2010.

A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems withapplications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1): 120–145, 2011.

G. B. Danzig and P. Wolfe. The decomposition algorithm for linear programming. *Econometrica*, 4:767–778, 1961.

M. D. Doan, T. Keviczky, and B. De Schutter. An iterative scheme for distributed model predictive control using Fenchel's duality. *Journal of Process Control*, 21(5): 746–755, June 2011. Special Issue on Hierarchical and Distributed Model Predictive Control.

A. Domahidi, A. Zgraggen, M.N. Zeilinger, M. Morari, and C.N. Jones. Efficient interior point methods for multistage problems arising in receding horizon control. In *51st IEEE Conference on Decision and Control*, pages 668–674, Maui, HI, USA, December 2012.

H. Everett. Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, 11:399–417, 1963.

H. J. Ferreau, H. G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit mpc. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.

E. Ghadimi, I. Shames, and M. Johansson. Multi-step gradient methods for networked optimization. *Signal Processing, IEEE Transactions on*, 61(21):5417–5429, Nov 2013.

P. Giselsson. A generalized distributed accelerated gradient method for DMPC with iteration complexity bounds. In *Proceedings of 2013 American Control Conference*, pages 327–333, Washington D.C., June 2013.

P. Giselsson. QPgen: A C code generator for quadratic optimization problems, 2014a. `http://www.control.lth.se/user/pontus.giselsson/qpgen/`.

P. Giselsson. Improved dual decomposition for distributed model predictive control. In *Proceedings of 2014 IFAC World Congress*, pages 1203–1209, Cape Town, South Africa, August 2014b.

P. Giselsson. Improved fast dual gradient methods for embedded model predictive control. In *Proceedings of 2014 IFAC World Congress*, pages 2303–2309, Cape Town, South Africa, August 2014c.

P. Giselsson and S. Boyd. Preconditioning in fast dual gradient methods. In *Proceedings of the 53rd Conference on Decision and Control*, pages 5040–5045, Los Angeles, CA, December 2014.

P. Giselsson, M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer. Accelerated gradient methods and dual decomposition in distributed model predictive control. *Automatica*, 49(3):829–833, 2013.

M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari. Multi-parametric toolbox 3.0, 2013.

R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, February 1990.

J. L. Jerez, P. J. Goulart, S. Richter, G. A. Constantinides, E. C. Kerrigan, and M. Morari. Embedded online optimization for model predictive control at megahertz rates. *IEEE Transactions on Automatic Control*, 2013. Submitted.

P. Kapasouris, M. Athans, and G. Stein. Design of feedback control systems for unstable plants with saturating actuators. In *Proceedings of the IFAC Symposium on Nonlinear Control System Design*, pages 302–307. Pergamon Press, 1990.

P. A. Knight, D. Ruiz, and B. Uçar. A Symmetry Preserving Algorithm for Matrix Scaling. Research Report RR-7552, INRIA, February 2011.

A. Kozma, E. Klintberg, S. Gros, and M. Diehl. An improved distributed dual newton-cg method for convex quadratic programming problems. In *Proceedings of 2014 American Control Conference*, 2014. Submitted.

M. Kraning, E. Chu, J. Lavaei, and S. Boyd. Dynamic network energy management via proximal message passing. *Foundations and Trends in Optimization*, 1(2):70–122, 2013.

Z. Lu and T. K. Pong. Minimizing condition number via convex programming. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1193–1211, 2011.

J. Mattingley and S. Boyd. Cvxgen: a code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012.

I. Necoara and V. Nedelcu. Rate analysis of inexact dual first-order methods application to dual decomposition. *IEEE Transactions on Automatic Control*, 59(5):1232–1243, May 2014.

I. Necoara and V. Nedelcu. On linear convergence of a distributed dual gradient algorithm for linearly constrained separable convex problems. *Automatica*, 2015. To appear.

I. Necoara and A. Patrascu. Duquad: an inexact (augmented) dual first order algorithm for quadratic programming. Available: `http://arxiv.org/abs/1504.05708`, March 2015.

Valentin Nedelcu, Ion Necoara, and Quoc Tran-Dinh. Computational complexity of inexact gradient augmented lagrangian methods: Application to constrained mpc. *SIAM Journal on Control and Optimization*, 52 (5):3109–3134, 2014.

R. R. Negenborn. *Multi-Agent Model Predictive Control with Applications to Power Networks*. PhD thesis, TU Delft, 2007.

Y. Nesterov. A method of solving a convex programming problem with convergence rate O $(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Netherlands, 1st edition, 2003.

Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, May 2005.

Yu. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

B. O'Donoghue, G. Stathopoulos, and S. Boyd. A splitting method for optimal control. *IEEE Transactions on Control Systems Technology*, 21(6):2432–2442, 2013.

P. Patrinos and A. Bemporad. An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Transactions on Automatic Control*, 59(1):18–33, 2014.

S. Richter, C. N. Jones, and M. Morari. Certification aspects of the fast gradient method for solving the dual of parametric convex programs. *Mathematical Methods of Operations Research*, 77(3):305–321, 2013.

R. T. Rockafellar. *Convex Analysis*, volume 28. Princeton Univercity Press, Princeton, NJ, 1970.

D. Ruiz. A scaling algorithm to equilibrate both rows and columns norms in matrices. Technical report, Rutherford Appleton Laboratories, 2001.

R. Sinkhorn and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. Technical report. Available: `http://www.csie.ntu.edu.tw/~b97058/tseng/papers/apgm.pdf`, May 2008.

F. Ullmann and S. Richter. FiOrdOs: Code generation for first order methods, 2012. `http://fiordos.ethz.ch/`.

C. Zalinescu. *Convex analysis in general vector spaces*. World Scientific, 2002.

W. Zuo and Z. Lin. A generalized accelerated proximal gradient approach for total-variation-based image restoration. *IEEE Transactions on Image Processing*, 20(10):2748–2759, October 2011.

Appendix A. PROOF TO PROPOSITION 10

The function $f - \frac{1}{2}\|\cdot\|_H^2$ is linear on $\mathcal{B}_r^n(x^\star(\bar{\mu}))$ if and only if $f(x) = \frac{1}{2}\|x\|_H^2 + \langle \xi, x\rangle_2 + \theta$ for some $\xi \in \mathbb{E}_{H^{-1}}$, $\theta \in \mathbb{R}$ and all $x \in \mathcal{B}_r^n(x^\star(\bar{\mu}))$. Since $x^\star(\bar{\mu}) \in \mathcal{B}_r^n(x^\star(\bar{\mu}))$, we have

$$\begin{aligned} x^\star(\bar{\mu}) &= \operatorname*{argmin}_x \left\{ \tfrac{1}{2}\|x\|_H^2 + \langle \xi, x\rangle_2 + \theta + \langle A^T\bar{\mu}, x\rangle_2 \right\} \\ &= -H^{-1}(\xi + A^T\bar{\mu}). \end{aligned}$$

Due to continuity of $H^{-1}A^T$ we also have $x^\star(\bar{\mu} + \mu_d) = -H^{-1}(\xi + A^T(\bar{\mu} + \mu_d)) \in \mathcal{B}_r(x^\star(\bar{\mu}))$ for sufficiently small $\mu_d \in \mathbb{R}^m$ pointing in any direction. Thus, there exists a ball $\mathcal{B}_\epsilon^m(\bar{\mu})$ such that for each $\mu \in \mathcal{B}_\epsilon^m(\bar{\mu})$

$$\begin{aligned} d(\mu) &= -\min_x \left\{ \tfrac{1}{2}\|x\|_H^2 + \langle \xi, x\rangle_2 + \theta + \langle A^T\mu, x\rangle_2 \right\} \\ &= \tfrac{1}{2}\|A^T\mu + \xi\|_{H^{-1}}^2 - \theta. \end{aligned}$$

That is, d is a quadratic with Hessian $AH^{-1}A^T$ on $\mathcal{B}_\epsilon^m(\bar{\mu})$. This implies that for any $L \not\succeq AH^{-1}A^T$, where exist $\nu, \mu \in \mathcal{B}_\epsilon^m(\bar{\mu})$ (since $\mathcal{B}_\epsilon^m(\bar{\mu})$ has non-empty interior) such that (11) does not hold. This concludes the proof.

Appendix B. PROOF TO PROPOSITION 11

Since H is positive definite on the null-space of A, the inverse in (13) exists, see (Boyd and Vandenberghe, 2004, p. 523). Thus $x^\star(\mu) = -M_{11}(A^T\mu + \xi) + M_{12}b$, where $x^\star$ is defined in (8). Further

$$\begin{aligned} d(\mu) &= \frac{1}{2}\mu^T A(2M_{11} - M_{11}HM_{11})A^T\mu + \zeta^T\mu + \theta \\ &= \frac{1}{2}\mu^T AM_{11}A^T\mu + \zeta^T\mu + \theta \end{aligned}$$

where $\zeta \in \mathbb{R}^{m+p}$ and $\theta \in \mathbb{R}$ collect the linear and constant terms, and where $M_{11}HM_{11} = M_{11}$ is used in the second equality. This identity follows from the upper left block of $\left[\begin{smallmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{smallmatrix}\right]\left[\begin{smallmatrix} H & B^T \\ B & 0 \end{smallmatrix}\right]\left[\begin{smallmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{smallmatrix}\right] = \left[\begin{smallmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{smallmatrix}\right]$ and using $M_{11}^T B = M_{11}B = BM_{11} = 0$, where $BM_{11} = 0$ follows from the lower left block of $\left[\begin{smallmatrix} H & B^T \\ B & 0 \end{smallmatrix}\right]\left[\begin{smallmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{smallmatrix}\right] = \left[\begin{smallmatrix} I & 0 \\ 0 & I \end{smallmatrix}\right]$. This implies that (12) holds with $L = AM_{11}A^T$ and obviously for any $L \succeq AM_{11}A^T$. Further, since d is a quadratic function with Hessian $AM_{11}A^T$, (12) is tight for $L = AM_{11}A^T$ for all $\mu, \nu \in \mathbb{E}_{K^{-1}}$.Thus, no $L \not\succeq AM_{11}A^T$ exists such that (12) holds for all $\mu, \nu \in \mathbb{E}_{K^{-1}}$. This concludes the proof.