Primal-Dual Decomposition by Operator Splitting and Applications to Image Deblurring[*]

Daniel O'Connor[†] and Lieven Vandenberghe[‡]

Abstract. We present primal-dual decomposition algorithms for convex optimization problems with cost functions $f(x) + g(Ax)$, where f and g have inexpensive proximal operators and A can be decomposed as a sum of two structured matrices. The methods are based on the Douglas–Rachford splitting algorithm applied to various splittings of the primal-dual optimality conditions. We discuss applications to image deblurring problems with nonquadratic data fidelity terms, different types of convex regularization, and simple convex constraints. In these applications, the primal-dual splitting approach allows us to handle general boundary conditions for the blurring operator. Numerical results indicate that the primal-dual splitting methods compare favorably with the alternating direction method of multipliers, the Douglas–Rachford algorithm applied to a reformulated primal problem, and the Chambolle–Pock primal-dual algorithm.

1. Introduction. We discuss primal-dual splitting methods for convex optimization problems of the form

$$(1) \qquad\qquad \text{minimize} \quad f(x) + g(Ax),$$

where f and g are closed convex functions with inexpensive proximal operators and A is a structured matrix. Specifically, we distinguish two types of matrix structure. In the simplest case, the structure in A makes it possible to solve equations with a coefficient matrix $I + \lambda A^T A$, where $\lambda > 0$, efficiently. In the second and more general case, A can be decomposed as a sum $A = B + C$ of two structured matrices, with the same meaning of "structure": linear equations with coefficients $I + \lambda B^T B$ and $I + \lambda C^T C$ can be solved efficiently. These assumptions are motivated by applications in image processing. A variety of image deblurring problems can be formulated as large convex optimization problems of the form (1), with f and g simple convex penalty functions or indicator functions of simple convex sets. The matrix A represents the blurring operation and the linear transformations used in regularization terms (see section 4). The first of the two types of structure mentioned above arises when

[†]Department of Mathematics, University of California Los Angeles, Los Angeles, CA 90095 (daniel.v.oconnor@gmail.com).

[‡]Electrical Engineering Department, University of California Los Angeles, Los Angeles, CA 90095 (vandenbe@ucla.edu).

periodic boundary conditions are used. In this case $A^T A$ can be diagonalized by multiplication with DFT matrices, so equations with coefficient matrix $I + \lambda A^T A$ are solved efficiently via fast Fourier transforms. The second type of structure arises when more realistic boundary conditions (for example, replicate boundary conditions) are used. In the sum $A = B + C$ the first term then represents the model assuming periodic boundary conditions; the second term is a sparse term added to correct for the nonperiodic boundary conditions. Linear equations with coefficients $I + \lambda B^T B$ and $I + \lambda C^T C$ can therefore be solved efficiently via the fast Fourier transform and sparse matrix factorization methods, respectively. However, these techniques are not easily combined in an efficient method for inverting $I + \lambda A^T A$.

The algorithms we propose use the Douglas–Rachford splitting method applied to systems of primal-dual optimality conditions. This primal-dual approach is interesting for several reasons. In contrast to purely primal or dual applications of the Douglas–Rachford method (such as the alternating direction method of multipliers (ADMM)), it does not require a reformulation of the problem and the introduction of a potentially large number of auxiliary variables or constraints. An advantage over forward-backward splitting algorithms or semi-implicit primal-dual methods (such as the Chambolle–Pock algorithm) is that the selection of suitable step sizes is easier and is not limited by the norms of the linear operators. We test the performance of the primal-dual splitting method on a set of image deblurring examples. In each experiment the primal-dual method is compared with three other methods: ADMM, the Douglas–Rachford method applied to a reformulated primal problem, and the Chambolle–Pock method. The results indicate that the convergence of the primal-dual approach is comparable to or better than the other methods, although all perform well with properly chosen algorithm parameters.

The paper is organized as follows. In section 2 we present some background on monotone operators and the Douglas–Rachford splitting method. In section 3 we derive several primal-dual splitting strategies for the optimality conditions of (1), under the different assumptions on the structure in A. In the second half of the paper the results are applied to image deblurring problems with various types of convex regularization and constraints. In section 4 we first describe the image deblurring problem and formulate it as a convex optimization problem of the form (1). We then present five examples. In section 5 the primal-dual splitting method is applied to constrained total variation (TV) deblurring problems with a nonquadratic fidelity term. In section 6 we discuss image restoration problems with a tight frame regularization. In section 7 we consider an application to image deblurring with a spatially varying, piecewise-constant blurring operator. The paper concludes with a summary of the main points in section 8.

2. Douglas–Rachford splitting algorithm. In this section we provide a brief review of monotone operators and the Douglas–Rachford splitting algorithm. More details can be found in [17, 1, 3, 39].

2.1. Monotone operators and resolvents. A multivalued or set-valued operator $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^n$ maps points $x \in \mathbb{R}^n$ to sets $\mathcal{F}(x) \subset \mathbb{R}^n$. The domain of the operator is $\mathbf{dom}\,\mathcal{F} = \{x \mid \mathcal{F}(x) \neq \emptyset\}$, and its graph is the set $\{(x, y) \mid x \in \mathbf{dom}\,\mathcal{F}, y \in \mathcal{F}(x)\}$. The operator is *monotone* if

$$(y - \hat{y})^T (x - \hat{x}) \geq 0 \qquad \forall x, \hat{x}, \ y \in \mathcal{F}(x), \ \hat{y} \in \mathcal{F}(\hat{x}).$$

A monotone operator is *maximal monotone* if its graph is not a strict subset of the graph of another monotone operator.

The operator $(I + \lambda \mathcal{F})^{-1}$, where $\lambda > 0$, is called the *resolvent* of the operator $\mathcal{F}$. A fundamental result states that if F is maximal monotone, then $(I + \lambda \mathcal{F})^{-1}(\hat{x})$ exists and is unique for all $\hat{x} \in \mathbb{R}^n$ [4, Proposition 2.2]. The value $x = (I + \lambda \mathcal{F})^{-1}(\hat{x})$ of the resolvent is the unique solution of the monotone inclusion

$$(2) \qquad \hat{x} \in x + \lambda \mathcal{F}(x).$$

The operators we will encounter in this paper are combinations of two elementary types of maximal monotone operators. The first is the subdifferential ∂f of a closed convex function f with nonempty domain. The second type is a skew-symmetric linear operator of the form

$$(3) \qquad \mathcal{F}(x, z) = \begin{bmatrix} 0 & C^T \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix}.$$

We now discuss the resolvents of these two types of operators in more detail.

Proximal operator. If $\mathcal{F} = \partial f$, with f a closed convex function, the resolvent is also called the *proximal operator* or *prox-operator* of f and written $\text{prox}_{\lambda f} = (I + \lambda \partial f)^{-1}$. The prox-operator $\text{prox}_{\lambda f}$ maps x to the unique solution of the optimization problem

$$(4) \qquad \text{minimize} \quad f(u) + \frac{1}{2\lambda} \|u - x\|^2$$

with variable u. (Throughout the paper $\| \cdot \|$ denotes the Euclidean norm.) For example, the prox-operator of the indicator function of a nonempty closed convex set is the Euclidean projection on the set.

The proximal operators of a function f and its conjugate f^* are related by the formula

$$(5) \qquad x = \text{prox}_{\lambda f}(x) + \lambda \text{prox}_{\lambda^{-1} f^*}(x/\lambda) = \text{prox}_{\lambda f}(x) + \text{prox}_{f^* \lambda}(x).$$

Here, $f^* \lambda$ denotes the *right scalar multiplication* of f^* with λ, defined as $(f^* \lambda)(x) = \lambda f^*(x/\lambda)$ [40, page 35]. It is easily verified that $(f^* \lambda) = (\lambda f)^*$ and $\text{prox}_{f^* \lambda}(x) = \lambda \text{prox}_{\lambda^{-1} f^*}(x/\lambda)$.

The decomposition (5) is known as the *Moreau decomposition* [35] and can be used to compute the proximal operator of a function from the proximal operator of its conjugate. For example, if f is a norm, then its conjugate $f^*(y) = \delta_{\mathcal{B}}(y)$ is the indicator function of the unit ball for the dual norm. The proximal operator of f can therefore be expressed in terms of the Euclidean projection $P_{\mathcal{B}}$ on $\mathcal{B}$:

$$(6) \qquad \text{prox}_{\lambda f}(x) = x - \lambda P_{\mathcal{B}}(x/\lambda) = x - P_{\lambda \mathcal{B}}(x).$$

For $f(x) = \|x\|_1$, this is the familiar soft-thresholding operation

$$\text{prox}_{\lambda f}(x)_k = \begin{cases} x_k - \lambda, & x_k > \lambda, \\ 0, & -\lambda \leq x_k \leq \lambda, \\ x_k + \lambda, & x_k \leq -\lambda. \end{cases}$$

We will also encounter the following pair of dual norms: for $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$,

$$
(7) \qquad \|(x, y)\|_{\text{iso}} = \sum_{k=1}^{n} (x_k^2 + y_k^2)^{1/2}, \qquad \|(x, y)\|_{\text{iso}*} = \max_{k=1,\dots,n} (x_k^2 + y_k^2)^{1/2}.
$$

(The subscript refers to the use of this norm to express isotropic two-dimensional (2D) TV; see section 4.) Here, the dual norm ball is a product of n 2D Euclidean norm balls. Using (6) the prox-operator $(u, v) = \text{prox}_{\lambda f}(x, y)$ of $f(x, y) = \|(x, y)\|_{\text{iso}}$ can be computed as $(u_k, v_k) = \alpha_k(x_k, y_k)$ for $k = 1, \dots, n$ with

$$
\alpha_k = 1 - \frac{\lambda}{(x_k^2 + y_k^2)^{1/2}} \quad \text{if } (x_k^2 + y_k^2)^{1/2} > \lambda, \qquad \alpha_k = 0 \quad \text{otherwise.}
$$

We refer the reader to [14, 12, 39] for extensive overviews of the calculus of proximal operators. Here, we mention one more property that will be useful later. In general, there is no simple formula for the prox-operator of $f(x) = g(Ax + b)$, given the prox-operator of g. However, it can be shown that if $AA^T = \alpha I$, with $\alpha > 0$, then

$$
(8) \qquad \text{prox}_f(x) = \frac{1}{\alpha} \left((\alpha I - A^T A)x + A^T (\text{prox}_{\alpha g}(Ax + b) - b) \right).
$$

Skew-symmetric linear operators. The resolvent of a monotone (i.e., positive semidefinite) linear operator $\mathcal{F}(x) = Ax$ is the matrix inverse $(I + \lambda A)^{-1}$. We note the following useful expressions for the resolvent of the skew-symmetric linear operator (3):

$$
(9) \qquad \begin{bmatrix} I & \lambda C^T \\ -\lambda C & I \end{bmatrix}^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} I \\ \lambda C \end{bmatrix} (I + \lambda^2 C^T C)^{-1} \begin{bmatrix} I \\ -\lambda C \end{bmatrix}^T
$$

$$
(10) \qquad = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -\lambda C^T \\ I \end{bmatrix} (I + \lambda^2 C C^T)^{-1} \begin{bmatrix} \lambda C^T \\ I \end{bmatrix}^T.
$$

This can be given a regularized least-squares interpretation. Let (x, z) be the value of the resolvent of $\mathcal{F}$:

$$
\begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} I & \lambda C^T \\ -\lambda C & I \end{bmatrix}^{-1} \begin{bmatrix} \hat{x} \\ \hat{z} \end{bmatrix}.
$$

Then, from (9), we see that $z = \hat{z} + \lambda C x$ and x is the solution of the least-squares problem

$$
\text{minimize} \quad \|\lambda C x + \hat{z}\|^2 + \|x - \hat{x}\|^2.
$$

Alternatively, from the second expression, $x = \hat{x} - \lambda C^T z$ and z is the solution of

$$
\text{minimize} \quad \|\lambda C^T z - \hat{x}\|^2 + \|z - \hat{z}\|^2.
$$

2.2. Douglas–Rachford splitting. The problem of finding a zero of a monotone operator $\mathcal{F}$, i.e., solving $0 \in \mathcal{F}(x)$, is called a monotone inclusion problem. It is easily seen, for example, from (2), that the zeros of a maximal monotone operator $\mathcal{F}$ are the fixed points of the resolvent of $\mathcal{F}$. The fixed point iteration

$$
x^k = (I + t\mathcal{F})^{-1} x^{k-1}, \quad k = 1, 2, \dots,
$$

is called the *proximal point algorithm* for solving the inclusion problem $0 \in \mathcal{F}(x)$ [42]. The proximal point algorithm converges under weak conditions (namely, $\mathcal{F}^{-1}(0) \neq \emptyset$ and maximal monotonicity of $\mathcal{F}$; see, for example, [17, Theorem 3]) but is useful in practice only when the resolvent evaluations are inexpensive. Often this is not the case, and one has to resort to algorithms based on operator splitting. A splitting algorithm decomposes the operator $\mathcal{F}$ as a sum $\mathcal{F} = \mathcal{A} + \mathcal{B}$ of two maximal monotone operators and requires only the resolvents of $\mathcal{A}$ or $\mathcal{B}$ or both.

A simple algorithm for solving $0 \in \mathcal{A}(x) + \mathcal{B}(x)$ is the Douglas–Rachford algorithm [33, 20, 17, 11]. The algorithm can be written in a number of equivalent forms, including

$$
\begin{align}
x^k &= (I + t\mathcal{A})^{-1}(z^{k-1}), \tag{11}\\
y^k &= (I + t\mathcal{B})^{-1}(2x^k - z^{k-1}), \tag{12}\\
z^k &= z^{k-1} + \rho(y^k - x^k). \tag{13}
\end{align}
$$

There are two algorithm parameters: a positive step size t and a relaxation parameter $\rho \in (0,2)$. It can be shown that if $\mathcal{A} + \mathcal{B}$ has a zero, then z^k converges to a limit of the form $x + tv$, with $0 \in \mathcal{A}(x) + \mathcal{B}(x)$ and $v \in \mathcal{A}(x) \cap (-\mathcal{B}(x))$. Therefore, $x^k = (I + t\mathcal{A})^{-1}(z^{k-1})$ converges to a solution x [17]. Note that if $\mathcal{B}$ is linear, then the vector $r^k = (1/t)(I + t\mathcal{B})(x^k - y^k)$ satisfies

$$
\begin{align}
r^k &= \frac{1}{t}((I + t\mathcal{B})(x^k) - 2x^k + z^{k-1})\\
&= \mathcal{B}(x^k) + \frac{1}{t}(z^{k-1} - x^k)\\
&\in \mathcal{B}(x^k) + \mathcal{A}(x^k)
\end{align}
$$

and $r^k \to 0$ because $x^k - y^k \to 0$. We can therefore use a stopping criterion of the form $\|r^k\| \leq \epsilon$. The Douglas–Rachford method is useful if the resolvents of $\mathcal{A}$ and $\mathcal{B}$ are inexpensive, compared to the resolvent of the sum $\mathcal{A} + \mathcal{B}$.

2.3. Alternating minimization. The Douglas–Rachford splitting method can be used to minimize a sum $f(x) + g(x)$ of two convex functions by taking $\mathcal{A} = \partial f$ and $\mathcal{B} = \partial g$. In this case, the resolvents in the algorithm (11)–(13) are prox-operators, $(I + t\mathcal{A})^{-1} = \mathrm{prox}_{tf}$ and $(I + t\mathcal{B})^{-1} = \mathrm{prox}_{tg}$, and the method can be viewed as alternating between the two minimization problems that define these prox-operators. Two well-known algorithms, Spingarn's method of partial inverses [45, 46] and the ADMM [22, 21], can be interpreted as applications of this idea.

Spingarn's method of partial inverses [45, 46] is a method for minimizing a convex function f over a subspace $\mathcal{V}$:

$$
\begin{align}
\text{minimize} \quad & f(x)\\
\text{subject to} \quad & x \in \mathcal{V}.
\end{align}
$$

Eckstein and Bertsekas [17] have shown that the method is equivalent to the Douglas–Rachford method for minimizing $f(x) + \delta_{\mathcal{V}}(x)$, where $\delta_{\mathcal{V}}$ is the indicator function of $\mathcal{V}$. Each iteration in the algorithm requires an evaluation of the prox-operator of f and a Euclidean projection on the subspace $\mathcal{V}$.

The ADMM [22, 21] is a dual decomposition method for problems of the form

$$\begin{aligned} \text{minimize} \quad & f(x) + g(y) \\ \text{subject to} \quad & Ax + By = c. \end{aligned}$$

The method is also known in image processing as the split Bregman method [23]. As shown in [20, 17], the algorithm can be interpreted as the Douglas–Rachford method applied to the dual problem

$$\text{maximize} \quad c^T z - f^*(A^T z) - g^*(B^T z),$$

with the dual objective split as a sum of two concave functions $c^T z - f^*(A^T z)$ and $-g^*(B^T z)$. After a series of simplifications, the Douglas–Rachford iteration (with $\rho = 1$) can be written as

$$x^k = \underset{x}{\operatorname{argmin}} \, L(x, y^{k-1}, z^{k-1}),$$

(14)
$$y^k = \underset{y}{\operatorname{argmin}} \, L(x^k, y, z^{k-1}),$$

(15)
$$z^k = z^{k-1} + t(Ax^k + By^k - c),$$

where L is the augmented Lagrangian

$$L(x, y, z) = f(x) + g(y) + z^T(Ax + By - c) + \frac{t}{2}\|Ax + By - c\|^2.$$

To improve convergence one can modify the basic ADMM algorithm to include overrelaxation [17, 3]. In the overrelaxed version of ADMM, we replace the expression Ax^k in (14) and (15) with $\rho Ax^k - (1 - \rho)(By^{k-1} - c)$, where $\rho \in (0, 2)$ [3, page 21]. For more details on ADMM we refer the reader to the recent surveys [3, 16].

3. Primal-dual splitting strategies. We now apply the Douglas–Rachford splitting method to develop decomposition algorithms for the optimization problem (1). This problem format is widely used as a standard form in the literature on multiplier and splitting methods (see, for example, [21, 20]). Its popularity derives from the fact that large-scale problems in practical applications can often be expressed in this form with relatively simple choices for f, g, and A.

Note that there is flexibility in the choice of g and A. Replacing the problem with

(16)
$$\text{minimize} \quad f(x) + \tilde{g}(\tilde{A}x),$$

where $\tilde{A} = \beta A$ and $\tilde{g}(y) = g(y/\beta)$, does not change the optimal solution x or the optimal value of the problem. This observation will allow us to introduce an additional algorithm parameter that can be adjusted to improve convergence.

3.1. Optimality conditions. The optimality conditions for (1) can be written as

(17)
$$0 \in \begin{bmatrix} 0 & 0 & A^T & I \\ 0 & 0 & -I & 0 \\ -A & I & 0 & 0 \\ -I & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ \partial g(y) \\ 0 \\ \partial f^*(w) \end{bmatrix}.$$

(The second term on the right denotes the set $\{0\} \times \partial g(y) \times \{0\} \times \partial f^*(w)$.) The variable y is an auxiliary variable in the reformulated primal problem

(18)
$$\begin{array}{ll} \text{minimize} & f(x) + g(y) \\ \text{subject to} & Ax - y = 0. \end{array}$$

The variable z is the dual multiplier for the constraint $Ax = y$. The variable w is a variable in the dual problem, written as

$$\begin{array}{ll} \text{maximize} & -f^*(w) - g^*(z) \\ \text{subject to} & A^T z + w = 0. \end{array}$$

Several equivalent reduced forms of the optimality conditions can be obtained by eliminating y (using the relation $(\partial g)^{-1} = \partial g^*$ between the subdifferential of a function and its conjugate), by eliminating w (using $(\partial f^*)^{-1} = \partial f$), or by eliminating y and w. The first two of these options lead to 3×3 block systems

(19)
$$0 \in \begin{bmatrix} 0 & A^T & I \\ -A & 0 & 0 \\ -I & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ z \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ \partial g^*(z) \\ \partial f^*(w) \end{bmatrix},$$

(20)
$$0 \in \begin{bmatrix} 0 & 0 & A^T \\ 0 & 0 & -I \\ -A & I & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} \partial f(x) \\ \partial g(y) \\ 0 \end{bmatrix}.$$

Eliminating y and w results in a 2×2 system

(21)
$$0 \in \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} \partial f(x) \\ \partial g^*(z) \end{bmatrix}.$$

3.2. Simple splitting. The most straightforward primal-dual splitting approach is to write the right-hand side of the 2×2 system (21) as a sum of the two monotone operators

$$\mathcal{A}(x, z) = \begin{bmatrix} \partial f(x) \\ \partial g^*(z) \end{bmatrix}, \qquad \mathcal{B}(x, z) = \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix}.$$

The resolvents of the two operators were given in section 2.1. The value $(x, z) = (I + t\mathcal{A})^{-1}(\hat{x}, \hat{z})$ of the resolvent of $\mathcal{A}$ is given by

$$x = \text{prox}_{tf}(\hat{x}), \qquad z = \text{prox}_{tg^*}(\hat{z}).$$

Using the Moreau decomposition the second proximal operator can be written in an alternate form as

$$z = \hat{z} - t\,\text{prox}_{t^{-1}g}(\hat{z}/t) = \hat{z} - \text{prox}_{gt}(\hat{z})$$

(where gt denotes right scalar multiplication of g). The resolvent of $\mathcal{B}$ is a linear mapping

$$\begin{aligned} (I + t\mathcal{B})^{-1} &= \begin{bmatrix} I & tA^T \\ -tA & I \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} I \\ tA \end{bmatrix} (I + t^2 A^T A)^{-1} \begin{bmatrix} I \\ -tA \end{bmatrix}^T. \end{aligned}$$

An iteration of the Douglas–Rachford algorithm applied to $\mathcal{A} + \mathcal{B}$ is therefore as follows:

$$(22) \qquad x^k = \mathrm{prox}_{tf}(p^{k-1}),$$

$$(23) \qquad z^k = \mathrm{prox}_{tg^*}(q^{k-1}),$$

$$(24) \qquad \begin{bmatrix} u^k \\ v^k \end{bmatrix} = \begin{bmatrix} I & tA^T \\ -tA & I \end{bmatrix}^{-1} \begin{bmatrix} 2x^k - p^{k-1} \\ 2z^k - q^{k-1} \end{bmatrix},$$

$$(25) \qquad p^k = p^{k-1} + \rho(u^k - x^k),$$

$$(26) \qquad q^k = q^{k-1} + \rho(v^k - z^k).$$

The simple splitting strategy is useful when the prox-operators of f and g are inexpensive, and the structure in A allows us to solve linear equations with coefficient $I + t^2 A^T A$ efficiently.

It is interesting to work out the differences when the same method is applied to the scaled problem (16), i.e., when g is replaced with $\tilde{g}(y) = g(y/\beta)$ and A with $\tilde{A} = \beta A$. After a few simplifications in notation, the Douglas–Rachford iteration for the scaled problem can be written in terms of the original g and A as follows:

$$x^k = \mathrm{prox}_{tf}(p^{k-1}),$$

$$\tilde{z}^k = \mathrm{prox}_{sg^*}(\tilde{q}^{k-1}),$$

$$\begin{bmatrix} u^k \\ \tilde{v}^k \end{bmatrix} = \begin{bmatrix} I & tA^T \\ -sA & I \end{bmatrix}^{-1} \begin{bmatrix} 2x^k - p^{k-1} \\ 2\tilde{z}^k - \tilde{q}^{k-1} \end{bmatrix},$$

$$p^k = p^{k-1} + \rho(u^k - x^k),$$

$$\tilde{q}^k = \tilde{q}^{k-1} + \rho(\tilde{v}^k - \tilde{z}^k).$$

The parameter s is given by $s = \beta^2 t$ and can be interpreted as a dual step size, which can be chosen arbitrarily and independently of the (primal) step size t.

Primal Douglas–Rachford splitting (Spingarn's method). The type of structure exploited in the primal-dual "simple splitting" method is also easily handled by the Douglas–Rachford method applied to the primal or the dual problem. In the primal application, we approach the reformulated problem (18) as one of minimizing a separable function $f(x) + g(y)$ over a subspace $\mathcal{V} = \{(x,y) \mid y = Ax\}$. The primal Douglas–Rachford method (or Spingarn's method) applied to this problem therefore involves evaluations of the prox-operator of $f(x) + g(y)$, i.e., the prox-operators of f and g, and Euclidean projection on $\mathcal{V}$. The projection (x,y) of a vector $(\hat{x}, \hat{y})$ on $\mathcal{V}$ can be expressed as

$$x = (I + A^T A)^{-1}(\hat{x} + A^T \hat{y}), \qquad y = Ax.$$

The steps in the primal Douglas–Rachford method are therefore very similar (but not equivalent) to (22)–(26). In particular, the complexity per iteration is the same.

Dual Douglas–Rachford splitting (ADMM). In the dual application of the Douglas–Rachford method, better known as ADMM, one first reformulates the problem by introducing a "dummy" variable u and adding a constraint $u = x$:

$$(27) \qquad \begin{array}{ll} \text{minimize} & f(u) + g(y) \\ \text{subject to} & \begin{bmatrix} I \\ A \end{bmatrix} x - \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix} = 0. \end{array}$$

The augmented Lagrangian of this problem is

$$f(u) + g(y) + w^T(x - u) + z^T(Ax - y) + \frac{t}{2}\left(\|x - u\|^2 + \|Ax - y\|^2\right).$$

In ADMM, one alternates between minimization of the augmented Lagrangian over x and over (u, y). Minimization over x is a least-squares problem with Hessian matrix $I + A^T A$. Minimization over u, y requires evaluations of the prox-operators of f and g. One iteration can be written as follows:

$$x^k = (I + A^T A)^{-1}\left(u^{k-1} + A^T y^{k-1} - \frac{1}{t}(w^{k-1} + A^T z^{k-1})\right),$$
$$u^k = \mathrm{prox}_{t^{-1}f}(x^k + w^{k-1}/t),$$
$$y^k = \mathrm{prox}_{t^{-1}g}(Ax^k + z^{k-1}/t),$$
$$w^k = w^{k-1} + t(x^k - u^k),$$
$$z^k = z^{k-1} + t(Ax^k - y^k).$$

As can be seen, the per-iteration complexity is the same as in the primal-dual and primal methods. One difference is that ADMM requires the introduction of an additional variable u. If the residual in the equality constraint $x = u$ is slow to decrease to zero, this can be expected to impact the convergence of x. We will return to this issue in section 3.5.

Chambolle–Pock method. Another interesting method for (1) is the Chambolle–Pock algorithm [6]. One iteration of the Chambolle–Pock method consists of the following steps:

$$x^k = \mathrm{prox}_{tf}(x^{k-1} - tA^T z^{k-1}),$$
$$z^k = \mathrm{prox}_{tg^*}(z^{k-1} + tA(2x^k - x^{k-1})).$$

This algorithm has the important advantage that it requires not the solution of linear equations but only multiplications with A and A^T. However, convergence of the algorithm depends on the step size t, which must be chosen in $(0, 1/\|A\|)$, where $\|A\|$ is the maximum singular value of A. As shown in [6], the method can be interpreted as a preconditioned ADMM. When one of the functions f or g^* is strongly convex, an accelerated version of Chambolle–Pock can be used. Since the strong convexity assumption does not hold for the applications studied in this paper, we omit the details.

Different primal and dual step sizes t and s can be used in the Chambolle–Pock method. With different step sizes the iteration is

$$x^k = \mathrm{prox}_{tf}(x^{k-1} - tA^T z^{k-1}),$$
$$z^k = \mathrm{prox}_{sg^*}(z^{k-1} + sA(2x^k - x^{k-1})).$$

Convergence is guaranteed if $\sqrt{st} < 1/\|A\|$. This variation of the Chambolle–Pock method can also be interpreted as the basic version of the method (with a single step size t) applied to the scaled problem (16). The dual step size s and the scale factor β are related by $s = \beta^2 t$. As a further improvement, one can apply overrelaxation (see [15] for details). The iteration

for the overrelaxed version is

$$\bar{x}^k = \text{prox}_{tf}(x^{k-1} - tA^T z^{k-1}),$$
$$\bar{z}^k = \text{prox}_{sg^*}(z^{k-1} + sA(2\bar{x}^k - x^{k-1})),$$
$$(x^k, z^k) = \rho(\bar{x}^k, \bar{z}^k) + (1 - \rho)(x^{k-1}, z^{k-1}),$$

where $\rho \in (0, 2)$.

3.3. Mixed splitting.

We now consider problem (1) under weaker assumptions on the problem structure. As before, we assume that f and g have inexpensive prox-operators. In addition we assume that A can be decomposed as $A = B+C$, where B and C have the property that equations with coefficients $I + \lambda B^T B$ and $I + \lambda C^T C$, with $\lambda > 0$, are easy to solve (but not necessarily equations with coefficient $I + \lambda A^T A$). We apply the Douglas–Rachford method to the primal-dual optimality condition (17), with the right-hand side decomposed as the sum of two monotone operators

$$\mathcal{A}(x, y, z, w) = \begin{bmatrix} 0 & 0 & B^T & 0 \\ 0 & 0 & 0 & 0 \\ -B & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ \partial g(y) \\ 0 \\ \partial f^*(w) \end{bmatrix},$$

$$\mathcal{B}(x, y, z, w) = \begin{bmatrix} 0 & 0 & C^T & I \\ 0 & 0 & -I & 0 \\ -C & I & 0 & 0 \\ -I & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}.$$

The resolvents of $\mathcal{A}$ and $\mathcal{B}$ can be evaluated as follows. The value $(x, y, z, w) = (I + t\mathcal{A})^{-1}(\hat{x}, \hat{y}, \hat{z}, \hat{w})$ of the resolvent of $\mathcal{A}$ is the solution of the inclusion problem

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ \hat{w} \end{bmatrix} \in \begin{bmatrix} I & 0 & tB^T & 0 \\ 0 & I & 0 & 0 \\ -tB & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} + t \begin{bmatrix} 0 \\ \partial g(y) \\ 0 \\ \partial f^*(w) \end{bmatrix}.$$

The computation of y, w, and (x, z) can be separated. The solution for y and w is

$$y = \text{prox}_{tg}(\hat{y}), \qquad w = \text{prox}_{tf^*}(\hat{w}).$$

The solution x, z follows from

$$\begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} I & tB^T \\ -tB & I \end{bmatrix}^{-1} \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} 0 \\ \hat{y} \end{bmatrix} + \begin{bmatrix} I \\ tB \end{bmatrix} (I + t^2 B^T B)^{-1}(\hat{x} - tB\hat{y})$$

(using the expression (9)). The resolvent of $\mathcal{B}$ is a linear operator. It can be verified that

$$
(I + t\mathcal{B})^{-1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{1+t^2}I & \frac{t}{1+t^2}I & 0 \\ 0 & -\frac{t}{1+t^2}I & \frac{1}{1+t^2}I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}
$$

$$
+ \begin{bmatrix} I \\ \frac{t^2}{1+t^2}C \\ \frac{t}{1+t^2}C \\ tI \end{bmatrix} \left((1+t^2)I + \frac{t^2}{1+t^2}C^TC \right)^{-1} \begin{bmatrix} I \\ \frac{t^2}{1+t^2}C \\ -\frac{t}{1+t^2}C \\ -tI \end{bmatrix}^T .
$$

From these expressions, we see that the resolvents of $\mathcal{A}$ and $\mathcal{B}$ require proximal operators of f and g, and linear equations with coefficients of the form $I + \lambda B^T B$ and $I + \lambda C^T C$. These operations are inexpensive, under our assumptions on the problem structure, and therefore each iteration of the Douglas–Rachford method is relatively cheap.

Primal Douglas–Rachford splitting (Spingarn's method). As one of the referees who reviewed this paper pointed out, the additive structure in $A = B + C$ can be handled in the primal Douglas–Rachford method, at the cost of doubling the number of variables, via the reformulation

$$
\text{(28)} \qquad \begin{array}{ll} \text{minimize} & f(x_1) + g(y_1 + y_2) + \delta(x_1 - x_2) \\ \text{subject to} & Bx_1 - y_1 = 0, \\ & Cx_2 - y_2 = 0, \end{array}
$$

where δ is the indicator function of $\{0\}$. The variables in the reformulated problem are x_1, x_2, y_1, y_2. The cost function and its proximal operator are separable in two sets of variables (x_1, x_2) and (y_1, y_2). The proximal operator of $f(x_1) + \delta(x_1 - x_2)$, as a function of x_1 and x_2, reduces to an evaluation of the prox-operator of f. The prox-operator of $g(y_1 + y_2)$ as a function of y_1 and y_2 can be related to the prox-operator of g via the formula (8). The Euclidean projection (x_1, x_2, y_1, y_2) of $(\hat{x}_1, \hat{x}_2, \hat{y}_1, \hat{y}_2)$ on the subspace defined by the constraints is given by

$$
x_1 = (I + B^T B)^{-1}(\hat{x}_1 + B^T \hat{y}_1), \qquad x_2 = (I + C^T C)^{-1}(\hat{x}_2 + C^T \hat{y}_2),
$$

and $y_1 = Bx_1$, $y_2 = Cx_2$.

Dual Douglas–Rachford splitting (ADMM). A similar reformulation, with an extra dummy variable u, brings the problem into a form amenable to ADMM:

$$
\begin{array}{ll} \text{minimize} & f(u) + g(y_1 + y_2) \\[2mm] \text{subject to} & \begin{bmatrix} I & 0 \\ 0 & I \\ B & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} I & 0 & 0 \\ I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} u \\ y_1 \\ y_2 \end{bmatrix} = 0. \end{array}
$$

One iteration of ADMM involves an alternating minimization of the augmented Lagrangian over (x_1, x_2), and (u, y_1, y_2). This requires evaluations of the prox-operators of f and g and the solution of least-squares problems with Hessian matrices $I + B^T B$ and $I + C^T C$.

Chambolle–Pock method. The Chambolle–Pock method described in section 3.2 involves only multiplications with A and A^T and therefore applies without modification when A can be decomposed as $A = B + C$ with structured B and C.

3.4. Mixed splitting with partially quadratic functions. The mixed splitting strategy can be simplified if f or g is a simple quadratic function, or a separable function that includes quadratic terms. We illustrate this with two examples.

Quadratic f. Suppose f in (1) is quadratic, of the form $f(x) = (\mu/2)x^T x + a^T x$ with $\mu \geq 0$. In that case we can start from the optimality conditions in the simpler 3×3 form (20). Since $\partial f(x) = \{\mu x + a\}$, we can use the splitting

$$\mathcal{A}(x, y, z) = \begin{bmatrix} \mu I & 0 & B^T \\ 0 & 0 & 0 \\ -B & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} a \\ \partial g(y) \\ 0 \end{bmatrix},$$

$$\mathcal{B}(x, y, z) = \begin{bmatrix} 0 & 0 & C^T \\ 0 & 0 & -I \\ -C & I & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

The resolvent of $\mathcal{A}$ maps $(\hat{x}, \hat{y}, \hat{z})$ to the solution (x, y, z) of the inclusion

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} \in \begin{bmatrix} (1 + \mu t)I & 0 & tB^T \\ 0 & I & 0 \\ -tB & 0 & I \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + t \begin{bmatrix} a \\ \partial g(y) \\ 0 \end{bmatrix}.$$

The solution is $y = \mathrm{prox}_{tg}(\hat{y})$ and

$$\begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} (1 + \mu t)I & tB^T \\ -tB & I \end{bmatrix}^{-1} \begin{bmatrix} \hat{x} - ta \\ \hat{z} \end{bmatrix}$$

$$= \left(\begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} I \\ tB \end{bmatrix} ((1 + \mu t)I + t^2 B^T B)^{-1} \begin{bmatrix} I \\ -tB \end{bmatrix}^T \right) \begin{bmatrix} \hat{x} - ta \\ \hat{z} \end{bmatrix}.$$

The operator $\mathcal{B}$ is linear so its resolvent is a matrix inverse:

$$(I + t\mathcal{B})^{-1} = \frac{1}{1 + t^2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & I & tI \\ 0 & -tI & I \end{bmatrix}$$

$$+ \begin{bmatrix} I \\ \frac{t^2}{1+t^2}C \\ \frac{t}{1+t^2}C \end{bmatrix} \left(I + \frac{t^2}{1 + t^2}C^T C \right)^{-1} \begin{bmatrix} I \\ \frac{t^2}{1+t^2}C \\ -\frac{t}{1+t^2}C \end{bmatrix}^T.$$

The Douglas–Rachford method applied to $\mathcal{A} + \mathcal{B}$ is therefore of interest under the same assumptions as in the general mixed splitting case of section 3.3: the function g has an inexpensive proximal operator, and the matrices B and C possess structures that allow us to solve linear equations with coefficients of the form $I + \lambda B^T B$ and $I + \lambda C^T C$ quickly. However, since the number of variables in the 3×3 system (20) is smaller, one can expect the Douglas–Rachford method to converge faster than for the general mixed splitting described in section 3.3.

Quadratic g. A similar simplification is possible when g is quadratic, of the form $g(y) = \|y - a\|^2/(2\mu)$ for $\mu > 0$. In this case we start from the optimality conditions (19) and note that $g^*(z) = (\mu/2)z^T z + a^T z$ and $\partial g^*(z) = \{\mu z + a\}$. We choose the splitting

$$
\mathcal{A}(x, z, w) = \begin{bmatrix} 0 & B^T & 0 \\ -B & \mu I & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ z \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ a \\ \partial f^*(w) \end{bmatrix},
$$

$$
\mathcal{B}(x, z, w) = \begin{bmatrix} 0 & C^T & I \\ -C & 0 & 0 \\ -I & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ z \\ w \end{bmatrix}.
$$

The resolvent of $\mathcal{A}$ maps $(\hat{x}, \hat{z}, \hat{w})$ to the solution of

$$
\begin{bmatrix} \hat{x} \\ \hat{z} \\ \hat{w} \end{bmatrix} \in \begin{bmatrix} I & tB^T & 0 \\ -tB & (1 + \mu t)I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} x \\ z \\ w \end{bmatrix} + t \begin{bmatrix} 0 \\ a \\ \partial f^*(w) \end{bmatrix}.
$$

The solution is $w = \mathrm{prox}_{tf^*}(\hat{w})$ and

$$
\begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} I & tB^T \\ -tB & (1 + \mu t)I \end{bmatrix}^{-1} \begin{bmatrix} \hat{x} \\ \hat{z} - ta \end{bmatrix}
$$

$$
= \left(\frac{1}{1 + \mu t} \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} I \\ \frac{t}{1+\mu t}B \end{bmatrix} \left(I + \frac{t^2}{1 + \mu t} B^T B \right)^{-1} \begin{bmatrix} I \\ -\frac{t}{1+\mu t}B \end{bmatrix}^T \right) \begin{bmatrix} \hat{x} \\ \hat{z} - ta \end{bmatrix}.
$$

The resolvent of $\mathcal{B}$ is

$$
(I + t\mathcal{B})^{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} + \begin{bmatrix} I \\ tC \\ tI \end{bmatrix} \left((1 + t^2)I + t^2 C^T C \right)^{-1} \begin{bmatrix} I \\ -tC \\ -tI \end{bmatrix}^T.
$$

Evaluating the two resolvents requires an evaluation of the proximal operator of f^*, a linear equation with coefficient of the form $I + \lambda B^T B$, and a linear equation $I + \lambda C^T C$.

Partially quadratic functions. More generally, one can simplify the general mixed splitting method whenever f is separable, of the form $f(x_1, x_2) = f_1(x_1) + f_2(x_2)$ with f_1 a simple quadratic term, or g is separable, $g(y_1, y_2) = g_1(y_1) + g_2(y_2)$ with g_1 quadratic. The details are straightforward, and the idea will be illustrated with an example in section 6.2.

3.5. Example. The main advantage of the primal-dual splitting method is that it exploits the problem structure without introducing extra variables and constraints. This can be expected to benefit the speed of convergence and accuracy attained. The primal and dual Douglas–Rachford methods, on the other hand, have the same complexity per iteration as the primal-dual method, but only after a reformulation of the problem which requires extra variables and constraints. We also note that the increase in problem dimensions is greater in the dual than in the primal approach. Unfortunately a theoretical analysis of this effect is lacking, and it is not clear how important it is in practice. To conclude this section we therefore give a small experiment that illustrates the difference.
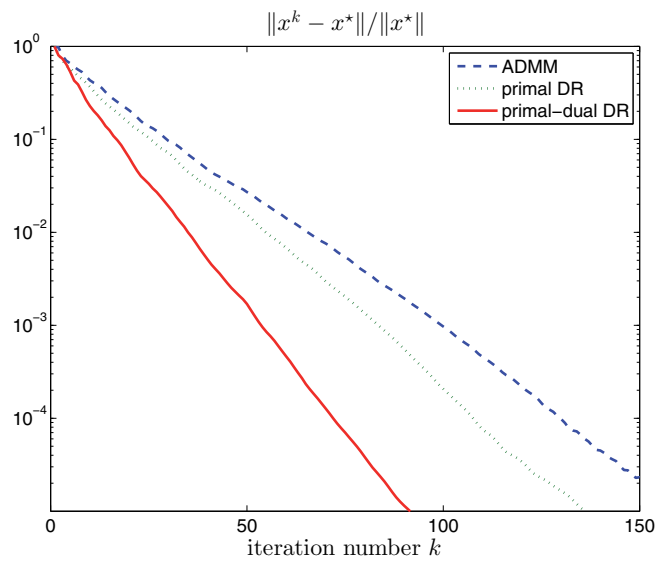
Figure 1. *Relative error versus iteration number for the experiment in section* 3.5.

We compare the mixed splitting approaches of section 3.3 on the test problem

$$\text{minimize} \quad \|x\| + \gamma\|(B+C)x - b\|_1$$

with $n = 500$ variables and square matrices B, C. This problem has the form (1) with $f(x) = \|x\|$ and $g(y) = \gamma\|y - b\|_1$, so the algorithms of section 3.3 can be applied. The problem data are generated randomly, with the components of A, B, b drawn independently from a standard normal distribution, $C = A - B$, and $\gamma = 1/100$. In Figure 1 we compare the convergence of the primal-dual mixed splitting method, the primal Douglas–Rachford method, and ADMM (dual Douglas–Rachford method). The relative error $\|x^k - x^\star\|/\|x^\star\|$ is with respect to the solution $x^\star$ computed using CVX [25, 24]. For each method, the three algorithm parameters (primal and dual step sizes and overrelaxation parameter) were tuned by trial and error to give fastest convergence. As can be seen, the primal-dual splitting method shows a clear advantage on this problem class. We also note that ADMM is slightly slower than the primal Douglas–Rachford method, which is consistent with the intuition that having fewer auxiliary variables and constraints is better.

4. Image deblurring by convex optimization. In the second half of the paper we apply the primal-dual splitting methods to image deblurring. We first discuss the blurring model and express the deblurring problem in a general optimization problem of the form (1). Let b be a vector containing the pixel intensities of an $N \times N$ blurry, noisy image, stored in column-major order as a vector of length $n = N^2$. Assume b is generated by a linear blurring operation with additive noise, i.e.,

$$(29) \qquad\qquad\qquad\qquad b = Kx_\text{t} + w,$$

where K is the blurring operator, $x_\text{t} \in \mathbb{R}^n$ is the unknown true image, and w is noise. The deblurring problem is to estimate x_t from b. Since blurring operators are often very ill-

conditioned the solution $x = K^{-1}b$ is a poor estimate of x_t, and regularization or constraints must be applied to obtain better estimates [26]. We will formulate the deblurring problem as an optimization problem of the following general form:

$$(30) \qquad\qquad \text{minimize} \quad \phi_\mathrm{f}(Kx - b) + \phi_\mathrm{r}(x) + \phi_\mathrm{s}(Dx),$$

with ϕ_f, ϕ_s, and ϕ_r convex penalty or indicator functions. This optimization problem is a special case of (1) with

$$f(x) = \phi_\mathrm{r}(x), \qquad g(u, v) = \phi_\mathrm{f}(u - b) + \phi_\mathrm{s}(v), \qquad A = \begin{bmatrix} K \\ D \end{bmatrix}.$$

We now discuss the three terms in (30) in more detail.

The first term in (30) is called the data fidelity term and penalizes or limits the deviation between the observed image and the model output for the reconstructed image. Typical choices for ϕ_f include a quadratic penalty $\phi_\mathrm{f}(u) = (1/2)\|u\|^2$ with $\|\cdot\|$ the Euclidean norm, the 1-norm penalty $\phi_\mathrm{f}(u) = \|u\|_1$, or the Huber penalty

$$\phi_\mathrm{f}(u) = \sum_{k=1}^{n} h_\eta(u_k), \qquad h_\eta(v) = \begin{cases} v^2/(2\eta), & |v| \le \eta, \\ |v| - \eta/2, & |v| \ge \eta. \end{cases}$$

One can also consider an indicator function $\phi_\mathrm{f}(u) = \delta_S(u)$ of a closed convex set S. For example, if S is a Euclidean norm ball $S = \{u \mid \|u\| \le \sigma\}$ and we take $\phi_\mathrm{f} = \delta_S$, the problem is equivalent to

$$\begin{aligned} \text{minimize} \quad & \phi_\mathrm{s}(Dx) + \phi_\mathrm{r}(x) \\ \text{subject to} \quad & \|Kx - b\| \le \sigma. \end{aligned}$$

More generally, ϕ_f can be a penalty function with a nontrivial domain, such as $\phi_\mathrm{f}(u) = -\sum_i \log(1 - u_i^2)$ with $\mathbf{dom}\,\phi_\mathrm{f} = \{u \mid \|u\|_\infty < 1\}$.

Practical algorithms must exploit the fact that the blurring matrix K is highly structured. For example, if K represents convolution with a spatially invariant point spread function and periodic boundary conditions are used, then it is block-circulant with circulant blocks. It therefore has a spectral decomposition $K = Q^H \operatorname{diag}(\lambda)Q$, where $Q = W \otimes W$ is the Kronecker product of the length-N DFT matrix with itself, and λ is the 2D DFT of the convolution kernel [26]. If the point spread function is also doubly symmetric and reflexive boundary conditions are used, then K can be diagonalized in a similar way by multiplication with orthogonal 2D discrete cosine transform (DCT) matrices [26, 37]. For other types of boundary conditions (for example, zero or replicate boundary conditions) K can be expressed as a sum $K = K_\mathrm{c} + K_\mathrm{s}$, where K_c is the blurring operator for periodic or reflexive boundary conditions, and K_s is a sparse term that corrects the values near the boundary.

The second term in (30) represents a regularization term or a constraint on x. Typical choices for ϕ_r are a quadratic penalty $\phi_\mathrm{r}(x) = \gamma\|x\|^2/2$, or the indicator function of a convex set (for example, box constraints $x \in [0, 1]^n$, added to limit the range of the pixel intensities). As pointed out in [48, 2, 7], the explicit addition of box or nonnegativity constraints on x can improve the quality of the restoration substantially.

The last term in (30) can be used to add a smoothing penalty. In that case the matrix $D \in \mathbb{R}^{2n \times n}$ is defined as concatenated vertical and horizontal discretized derivative operators. With periodic boundary conditions, D is given by

$$D = \begin{bmatrix} I \otimes D_1 \\ D_1 \otimes I \end{bmatrix}, \qquad D_1 = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ 0 & 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \\ 1 & 0 & 0 & \cdots & 0 & -1 \end{bmatrix} \in \mathbb{R}^{n \times n}.$$

If we choose for $\phi_{\mathrm{s}} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ the norm

$$\phi_{\mathrm{s}}(u, v) = \gamma \|(u, v)\|_{\mathrm{iso}} = \gamma \sum_{k=1}^{n} \sqrt{u_k^2 + v_k^2},$$

then $\phi_{\mathrm{s}}(Dx) = \gamma \, \mathrm{TV}(x)$ is the TV penalty introduced by Rudin, Osher, and Fatemi [43]. For quadratic ϕ_{f} and zero ϕ_{r} the general problem (30) then reduces to the classic TV-regularized deblurring problem

$$(31) \qquad\qquad \text{minimize} \quad \frac{1}{2}\|Kx - b\|^2 + \gamma \, \mathrm{TV}(x).$$

Another interesting choice for the third term in (30) is $\phi_{\mathrm{s}}(Dx) = \|Dx\|_1$, where D represents a wavelet or shearlet transform matrix.

In summary, the problem format (30) includes a variety of useful formulations of the image deblurring problem. In all of these examples the functions ϕ_{f}, ϕ_{r}, and ϕ_{s} are convex and relatively simple (i.e., have inexpensive prox-operators), but they are not necessarily quadratic or differentiable, and they can have a restricted domain. When the functions ϕ_{f}, ϕ_{r}, ϕ_{s} are quadratic (squared Euclidean norms), the problem reduces to a linear equation of the form

$$(K^T K + \rho I + \sigma D^T D)x = K^T b.$$

If the matrices K and D can be diagonalized by multiplication with DFT or DCT matrices, fast Fourier transform methods can be used to solve the equation in $O(N^2 \log N)$ operations [48, 26].

When one or more of the terms in the objective is not quadratic, the problem must be solved by an iterative optimization algorithm, customized to exploit the structure in K and D. Second-order methods, such as Newton's method or interior-point methods [8], [48, chapter 8], are limited because they require the solution of linear equations $K^T H_{\mathrm{f}} K + H_{\mathrm{r}} + D^T H_{\mathrm{s}} D$, where the matrices H_{f}, H_{r}, H_{s} are multiples of the Hessians of ϕ_{f}, ϕ_{r}, ϕ_{s} (if these functions are twice differentiable) or positive definite matrices that result from block-elimination of the Newton equations in an interior-point method. The presence of these matrices makes it difficult to exploit structure in K and D. Iterative linear equation solvers can be applied, using the fast Fourier transform for matrix-vector multiplications, but the reduced accuracy

of iterative equation solvers often impairs the fast convergence of Newton's method or the interior-point method.

Most published deblurring algorithms were developed for special cases of problem (30), such as the TV-regularized deblurring problem (31). Several of the existing algorithms of TV-regularized deblurring can be generalized in a straightforward manner to the general problem. Two such classes of methods are the ADMM and Chambolle–Pock algorithms introduced in the previous section. The ADMM or split Bregman method described in section 3.2 is a popular dual decomposition algorithm for large-scale convex optimization [22, 21, 20, 23, 3, 7]. As illustrated in [3], it can be used to derive simple, efficient algorithms for a range of applications. To apply ADMM to problem (30) (in its most general case, with nonquadratic ϕ_{f}, ϕ_{r}, ϕ_{s}), one first rewrites the problem in a form suited for dual decomposition, i.e., with a separable objective and linear equality constraints,

$$
\begin{aligned}
\text{minimize} \quad & \phi_{\mathrm{f}}(v) + \phi_{\mathrm{r}}(u) + \phi_{\mathrm{s}}(w) \\
\text{subject to} \quad & u = x, \\
& v = Kx - b, \\
& w = Dx,
\end{aligned}
$$

where u, v, w are auxiliary variables. The main step in each ADMM iteration is an alternating minimization of the augmented Lagrangian for this problem,

$$
\begin{aligned}
L(x, u, v, w) = {}& \phi_{\mathrm{f}}(v) + \phi_{\mathrm{r}}(u) + \phi_{\mathrm{s}}(w) \\
& + \frac{t}{2} \left(\left\| u - x + \frac{1}{t}p \right\|^2 + \left\| v - Kx + b + \frac{1}{t}q \right\|^2 + \left\| w - Dx + \frac{1}{t}r \right\|^2 \right).
\end{aligned}
$$

In this expression t is a positive step size, and p, q, and r denote multipliers for the three constraints. The first of the two alternating minimization steps is over x, keeping u, v, w constant; the second is over (u, v, w) with fixed x. Since L is separable in u, v, w, the second step involves three independent minimizations. The minimizations over u, v, w are straightforward (often with complexity $O(N^2)$) if the functions ϕ_{f}, ϕ_{f}, and ϕ_{s} are simple, as will be the case in the applications considered here. The minimization over x involves a linear equation with coefficient matrix $I + K^T K + D^T D$. If K and D are diagonalizable by multiplication with DFT or DCT matrices, this can be reduced to a diagonal equation and solved in $O(N^2 \log N)$ operations. The total cost of one iteration of the ADMM method is therefore $O(N^2 \log N)$.

A second important and versatile class of first-order algorithms contains the Chambolle–Pock algorithm and the related primal-dual methods developed in [18, 19, 5, 27, 15]. These methods solve the primal-dual optimality conditions

$$
(32) \qquad 0 \in \begin{bmatrix} 0 & K^T & D^T \\ -K & 0 & 0 \\ -D & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} \partial\phi_{\mathrm{r}}(x) \\ b + \partial\phi_{\mathrm{f}}^*(y) \\ \partial\phi_{\mathrm{s}}^*(z) \end{bmatrix}
$$

by a semi-implicit forward-backward iteration which requires matrix-vector multiplications with K and D and their transposes but not the solution of any linear equations. Several other

types of modified forward-backward splitting algorithms can be applied to the primal-dual optimality conditions, with a complexity per iteration similar to that of the Chambolle–Pock method [13, 47].

5. TV deblurring. In this section we present two applications of the primal-dual splitting methods to a constrained L_1 TV deblurring problem

$$(33) \qquad \begin{aligned} \text{minimize} \quad & \|Kx - b\|_1 + \gamma\|Dx\|_{\text{iso}} \\ \text{subject to} \quad & 0 \le x \le \mathbf{1}, \end{aligned}$$

where b is a blurry, noisy image with n pixels stored as a vector, K represents a convolution operator, and D represents a discrete gradient operator. The variable is an n-vector x. This problem can be written in the canonical form (1) with

$$(34) \qquad f(x) = \delta_S(x), \qquad g(y_1, y_2) = \|y_1 - b\|_1 + \gamma\|y_2\|_{\text{iso}}, \qquad A = \begin{bmatrix} K \\ D \end{bmatrix},$$

where $S = \{x \mid 0 \le x \le \mathbf{1}\}$.

All the experiments were performed on a computer with a 3.00 GHz AMD Phenom(tm) II X4 945 processor with 4 cores and 8 GB of RAM. The code was written in MATLAB using MATLAB version 8.1.0.604 (R2013a).

5.1. Periodic boundary conditions. In the first experiment, periodic boundary conditions are used in the definitions of K and D. The matrices $K^T K$ and $D^T D$ can therefore be diagonalized by the discrete Fourier basis matrix, and equations with coefficient

$$I + t^2 A^T A = I + t^2 K^T K + t^2 D^T D$$

are solved very efficiently via fast Fourier transforms. We can therefore apply the algorithms given in section 3.2.

Figure 2 compares the performances of the Chambolle–Pock, ADMM, primal Douglas–Rachford, and primal-dual Douglas–Rachford algorithms. The test image b is a degraded version of the 1024 by 1024 "Man" image from the USC-SIPI Image Database.[1] The original image (scaled so that intensity values are between 0 and 1) was blurred with a 15 by 15 truncated Gaussian kernel with standard deviation $\sigma = 7$. Then salt and pepper noise was added to a random selection of 50% of the pixels. The parameter γ was chosen to give a visually appealing image reconstruction. A nearly optimal primal objective value $f^\star$ was computed by running the primal-dual Douglas–Rachford algorithm for $10,000$ iterations. In Figure 2 the quantity $(f^k - f^\star)/f^\star$ is plotted against the iteration number k, where f^k is the primal objective value at iteration k. The original, blurry/noisy, and restored (by primal-dual Douglas–Rachford) images are shown in Figures 3 and 4.

For each method, close to optimal fixed primal and dual step sizes (and overrelaxation parameters) were selected by trial and error. The Chambolle–Pock step sizes s and t were chosen to satisfy $tsL^2 = 1$, where $L = (\|K\|^2 + \|D\|^2)^{1/2}$ is an upper bound on $\|A\|$. Note that the norms of K and D can be computed analytically because $K^T K$ and $D^T D$ are diagonalized
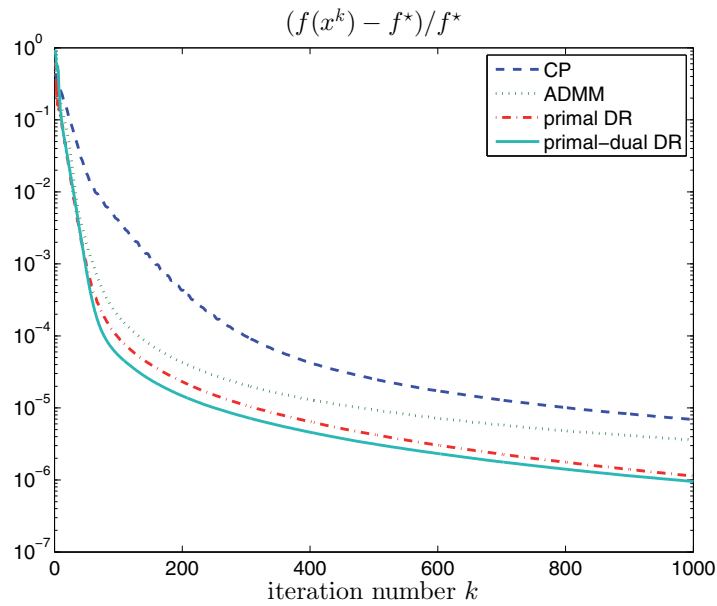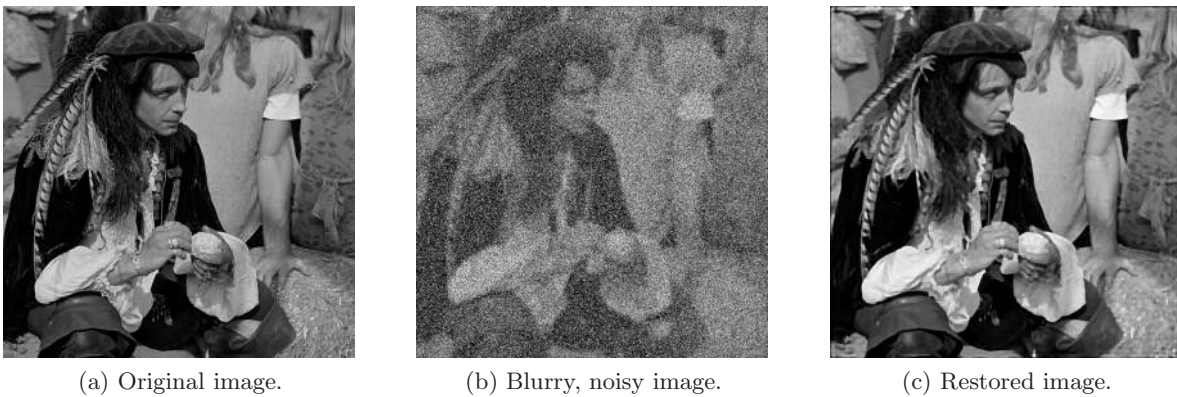
$$(f(x^k) - f^\star)/f^\star$$

Figure 2. *Relative optimality gap versus iteration number for the experiment in section* 5.1.

(a) Original image.   (b) Blurry, noisy image.   (c) Restored image.

**Figure 3.** *Result for the experiment in section* 5.1.

by the discrete Fourier basis matrix. The average elapsed time per iteration was 1.37 seconds for Chambolle–Pock, 1.33 seconds for ADMM, 1.33 seconds for primal Douglas–Rachford, and 1.46 seconds for primal-dual Douglas–Rachford.

As can be seen from the convergence plots, the four methods reach a modest accuracy quickly. After a few hundred iterations, progress slows down considerably. In this example the algorithms based on Douglas–Rachford converge faster than the Chambolle–Pock algorithm. The time per iteration is roughly the same for each method and is dominated by 2D fast Fourier transforms.

The quality of the restored image is good because the L_1 data fidelity is very well suited to deal with salt and pepper noise. Using an L_2 data fidelity term and omitting the interval constraints leads to a much poorer result. To illustrate this, Figure 5 shows the result of

(a) Close-up on original image.

(b) Close-up on degraded image.

(c) Close-up on restored image.
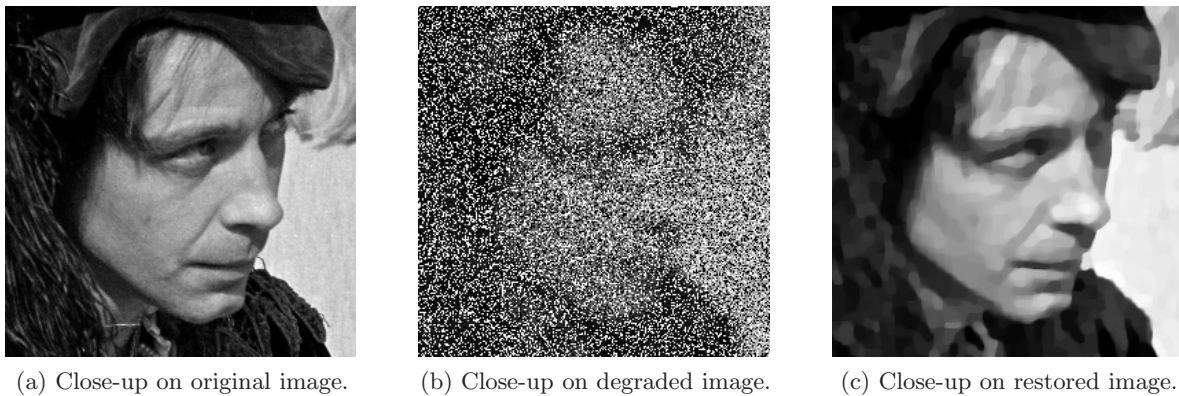
Figure 4. *Close-ups of the results for the experiment in section* 5.1.

(a) Deblurred using L_2 data fidelity term.

(b) Close-up on restoration using L_2 data fidelity term.
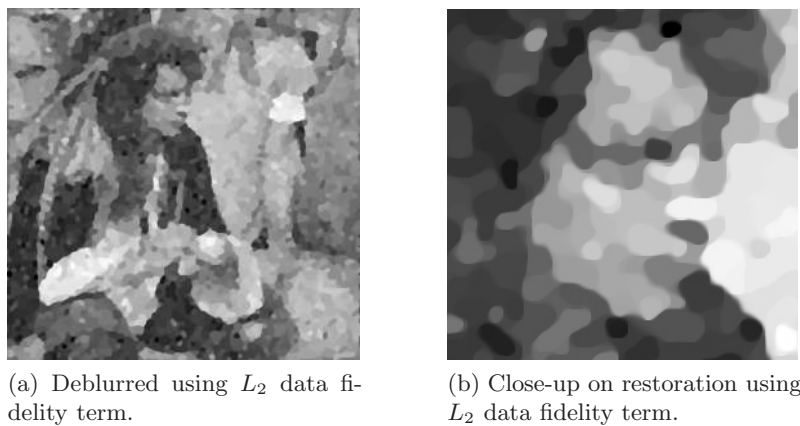
Figure 5. *Result of using L_2 data fidelity term for salt and pepper noise.*

minimizing $\|Kx - b\|^2 + \gamma\|Dx\|_{\mathrm{iso}}$, with γ chosen to give the most visually appealing reconstruction.

5.2. Nonperiodic boundary conditions. To illustrate the methods of section 3.3, we consider the same problem (33) with nonperiodic boundary conditions [9, 44]. Now K represents a convolution operator that uses replicate boundary conditions, and D represents a discrete gradient operator that uses symmetric boundary conditions. The matrices K and D can be decomposed as $K = K_{\mathrm{p}} + K_{\mathrm{s}}$ and $D = D_{\mathrm{p}} + D_{\mathrm{s}}$, where K_{p} and D_{p} represent convolution operators that use periodic boundary conditions, and K_{s} and D_{s} are sparse matrices that correct the values near the boundary. Correspondingly, the matrix A in (34) can be decomposed as $A = B + C$, where

$$B = \begin{bmatrix} K_{\mathrm{p}} \\ D_{\mathrm{p}} \end{bmatrix}, \qquad C = \begin{bmatrix} K_{\mathrm{s}} \\ D_{\mathrm{s}} \end{bmatrix}.$$

The two resolvents needed in the mixed splitting algorithm of section 3.3 are inexpensive to evaluate: in the resolvent of $\mathcal{A}$ one exploits the fact that $B^T B = K_{\mathrm{p}}^T K_{\mathrm{p}} + D_{\mathrm{p}}^T D_{\mathrm{p}}$ can be diagonalized by the DFT basis; the resolvent of $\mathcal{B}$ involves a linear equation with a sparse
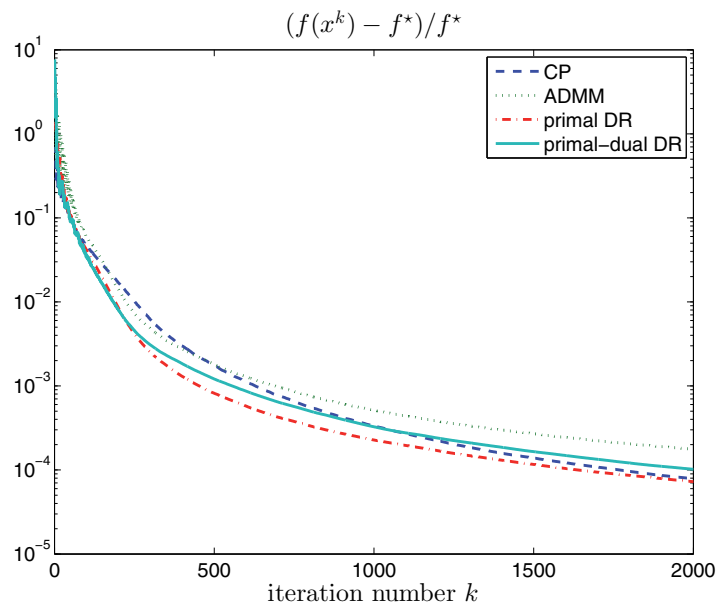
$$(f(x^k) - f^\star)/f^\star$$

Figure 6. *Relative optimality gap versus iteration for the experiment in section* 5.2.

(a) Blurred with 9 by 9 kernel, 10% salt and pepper noise.

(b) Restored using periodic boundary conditions.

(c) Restored using replicate boundary conditions.

Figure 7. *Result for the experiment in section* 5.2.

matrix $C^T C = K_s^T K_s + D_s^T D_s$.

Figure 6 compares the performances of the Chambolle–Pock, ADMM, primal Douglas–Rachford, and primal-dual Douglas–Rachford algorithms when b is a degraded version of the 256 by 256 "cameraman" image. The original image (scaled so that intensity values are between 0 and 1) was blurred with a 9 by 9 truncated Gaussian kernel with standard deviation $\sigma = 4$. Then salt and pepper noise was added to a random selection of 10% of the pixels. The parameter γ was chosen to give a visually appealing image reconstruction. A nearly optimal primal objective value $f^\star$ was computed by running the primal-dual Douglas–Rachford algorithm for 10,000 iterations. In Figure 6 the quantity $(f^k - f^\star)/f^\star$ is plotted against the iteration number k. (f^k is the primal objective value at iteration k.) The blurry/noisy and

restored (by primal-dual Douglas–Rachford) images are shown in Figure 7.

Also shown in Figure 7 is a restoration (by primal-dual Douglas–Rachford) using periodic boundary conditions (and all other parameters unchanged). In this example we see the value of using correct boundary conditions, as the quality of the restoration with incorrect boundary conditions is poor.

For all methods, close to optimal fixed primal and dual step sizes (and overrelaxation parameters) were selected by trial and error. Primal and dual step sizes were implemented in the algorithms based on Douglas–Rachford by modifying g and A as in (16). The Chambolle–Pock step sizes s and t were chosen to satisfy $tsL^2 = 1$, where $L = (\|K_\mathrm{p}\|^2 + \|D_\mathrm{p}\|^2)^{1/2} \approx \|A\|$. The norms of K_p and D_p can be computed analytically because $K_\mathrm{p}^T K_\mathrm{p}$ and $D_\mathrm{p}^T D_\mathrm{p}$ are diagonalized by the discrete Fourier basis. The average elapsed time per iteration was 0.08 seconds for Chambolle–Pock, 0.12 seconds for ADMM, 0.10 seconds for primal Douglas–Rachford, and 0.10 seconds for primal-dual Douglas–Rachford. In this example, the modification to Chambolle–Pock (incorporating an overrelaxation step) mentioned in section 3 allows Chambolle–Pock to be competitive with the algorithms based on Douglas–Rachford.

6. Tight frame regularized deblurring. In the next two experiments, we use tight frame regularization rather than TV regularization in the deblurring model (30). We take $\phi_\mathrm{s}(Dx) = \gamma\|Dx\|_1$, with D a tight frame analysis operator, i.e., $D^T D = \alpha I$ for some positive α [34, Chapter 5], [30].

6.1. Periodic boundary conditions. We first consider

$$
(35) \qquad \begin{array}{ll} \text{minimize} & \|Kx - b\|_1 + \gamma\|Dx\|_1 \\ \text{subject to} & 0 \leq x \leq \mathbf{1}, \end{array}
$$

where, as before, b is a blurry, noisy image stored as an n-vector and K represents a convolution operator constructed using periodic boundary conditions. The matrix D is the analysis operator for a (shearlet) tight frame [30]. The MATLAB package Shearlab-1.1 is used to evaluate the tight operator [31].

The problem (35) is a special case of problem (1) with

$$
f(x) = \delta_S(x), \qquad g(y_1, y_2) = \|y_1 - b\|_1 + \gamma\|y_2\|_1, \qquad A = \begin{bmatrix} K \\ D \end{bmatrix},
$$

and $S = \{x \mid 0 \leq x \leq \mathbf{1}\}$. Because $K^T K$ and $D^T D = \alpha I$ are both diagonalized by the discrete Fourier basis, we can use the simple splitting algorithm given in section 3.2.

Figure 8 compares the Chambolle–Pock, primal Douglas–Rachford, ADMM, and primal-dual Douglas–Rachford algorithms. The test image b is a degraded version of the 256 by 256 "cameraman" image. The original image (scaled so that intensity values are between 0 and 1) was blurred with a 7 by 7 truncated Gaussian kernel with standard deviation $\sigma = 5$. Then salt and pepper noise was added to a random selection of 30% of the pixels. The parameter γ was chosen to give a visually appealing image reconstruction. A nearly optimal primal objective value $f^\star$ was computed by running the primal-dual Douglas–Rachford algorithm for $10,000$ iterations. In Figure 8 the quantity $(f^k - f^\star)/f^\star$ is plotted against the iteration number k, where f^k is the primal objective value at iteration k. The original, blurry/noisy, and restored (by primal-dual splitting) images are shown in Figure 9.
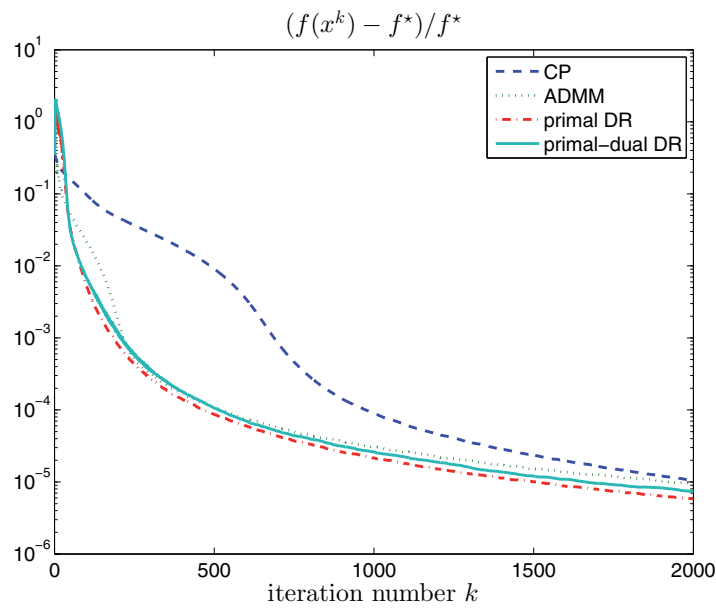
Figure 8. *Relative optimality gap versus iteration for the experiment in section* 6.1.

(a) Original image.   (b) Blurry, noisy image.   (c) Restored image.

Figure 9. *Result for the experiment in section* 6.1.

For each method, close to optimal fixed primal and dual step sizes (and overrelaxation parameters) were selected by trial and error. The Chambolle–Pock step sizes s and t were chosen to satisfy $tsL^2 = 1$, where $L = (\|K\|^2 + \|D\|^2)^{1/2}$ is an upper bound on $\|A\|$. Note that the norm of K can be computed analytically because $K^T K$ is diagonalized by the discrete Fourier basis matrix. The norm of D is $\sqrt{\alpha}$ because $D^T D = \alpha I$. The average elapsed time per iteration was 0.54 seconds for Chambolle–Pock, 0.51 seconds for ADMM, 0.51 seconds for primal Douglas–Rachford, and 0.50 seconds for primal-dual Douglas–Rachford.

The convergence plots are similar to those in the previous experiment (in section 5.1). As is typical for first-order methods, a modest accuracy is reached quickly, but progress slows down after the first few hundred iterations. In this example the algorithms based on Douglas–Rachford converge faster than the Chambolle–Pock algorithm. The time per iteration is

roughly the same for each method and is dominated by 2D fast Fourier transforms and shearlet transforms.

6.2. Nonperiodic boundary conditions.
We solve a problem similar to (35), but with a quadratic fidelity term and without the constraints on x:

$$(36) \qquad\qquad \text{minimize} \quad \frac{1}{2}\|Kx - b\|^2 + \gamma\|Dx\|_1.$$

We will use replicate boundary conditions for K. As in the previous experiment, D represents the analysis operator for a (shearlet) tight frame, constructed as in the previous example. The matrix K can be decomposed as $K = K_{\mathrm{p}} + K_{\mathrm{s}}$, where K_{p} represents a convolution operator that uses periodic boundary conditions. Therefore, $K_{\mathrm{p}}^T K_{\mathrm{p}}$ and $D^T D = \alpha I$ are both diagonalized by the discrete Fourier basis matrix. The matrix K_{s} is a sparse matrix that modifies the values near the boundary to satisfy the replicate boundary conditions.

This problem has the canonical form (1) with

$$f(x) = 0, \qquad g(y_1, y_2) = \frac{1}{2}\|y_1 - b\|^2 + \gamma\|y_2\|_1, \qquad A = \begin{bmatrix} K \\ D \end{bmatrix}.$$

Since f is quadratic, the simplified mixed splitting algorithm of section 3.4 can be applied. However, as indicated at the end of section 3.4, additional simplifications are possible because g is a separable sum of a quadratic term and a nonquadratic term.

Define $g_1(y_1) = \|y_1 - b\|^2/2$ and $g_2(y_2) = \gamma\|y_2\|_1$. The primal-dual optimality conditions for (36) can be written as

$$0 \in \begin{bmatrix} 0 & K^T & D^T \\ -K & 0 & 0 \\ -D & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \nabla g_1^*(z_1) \\ \partial g_2^*(z_2) \end{bmatrix}.$$

(Note that $g_1^*(z_1) = \|z_1\|^2/2 + b^T z_1$ and $\partial g_1^*(z_1) = \{\nabla g_1^*(z_1)\} = \{z_1 + b\}$.) We use the splitting

$$\mathcal{A}(x, z) = \begin{bmatrix} 0 & K_{\mathrm{s}}^T & 0 \\ -K_{\mathrm{s}} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \partial g_2^*(z_2) \end{bmatrix}$$

and

$$\mathcal{B}(x, z) = \begin{bmatrix} 0 & K_{\mathrm{p}}^T & D^T \\ -K_{\mathrm{p}} & I & 0 \\ -D & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix}.$$

Evaluating $(x, z_1, z_2) = (I + t\mathcal{A})^{-1}(\hat{x}, \hat{z}_1, \hat{z}_2)$ separates into two independent calculations: the solution of the linear equation

$$\begin{bmatrix} I & tK_{\mathrm{s}}^T \\ -tK_{\mathrm{s}} & I \end{bmatrix} \begin{bmatrix} x \\ z_1 \end{bmatrix} = \begin{bmatrix} \hat{x} \\ \hat{z}_1 \end{bmatrix}$$

and the evaluation of a prox-operator $z_2 = \mathrm{prox}_{tg_2^*}(\hat{z}_2)$. Evaluating the resolvent of $\mathcal{B}$ requires only solving a linear system. This system can be reduced to a system with a coefficient
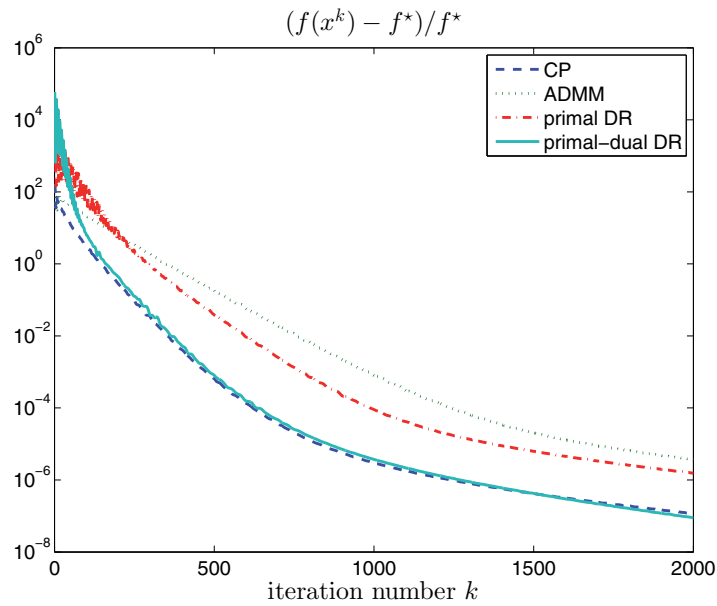
Figure 10. *Relative optimality gap versus iteration for the experiment in section* 6.2.

(a) Original image. (b) Blurry, noisy image. (c) Restored image.
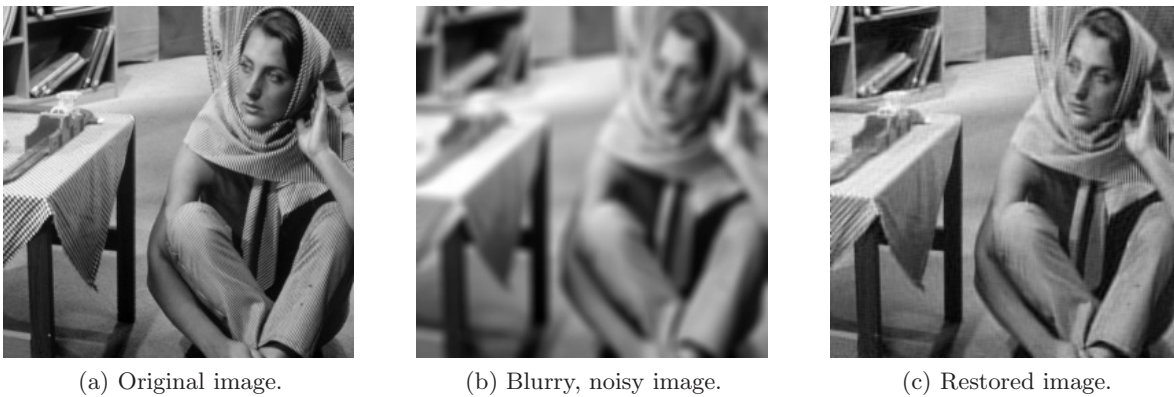
Figure 11. *Result for the experiment in section* 6.2.

consisting of positive weighted sum of three terms: $I + \lambda K_p^T K_p + \nu D^T D$. Hence the resolvent of $\mathcal{B}$ can be evaluated efficiently via fast Fourier transforms. Notice that we removed the variable y which appeared in the method of section 3.4 for quadratic f. This reduces the size of the monotone inclusion problem significantly.

Figure 10 compares the performances of the Chambolle–Pock, ADMM, primal Douglas–Rachford, and primal-dual Douglas–Rachford algorithms. The image b is a degraded version of the "Barbara" image, resized to have size 256 by 256. The original image (scaled so that intensity values are between 0 and 1) was blurred with a 9 by 9 truncated Gaussian kernel with standard deviation $\sigma = 4$. Then Gaussian noise with zero mean and standard deviation 10^{-3} was added to each pixel of the blurred image. The parameter γ was chosen to give a visually appealing image reconstruction. A nearly optimal primal objective value $f^\star$ was

computed by running the primal-dual Douglas–Rachford algorithm for $10,000$ iterations. In Figure 10 the quantity $(f^k - f^\star)/f^\star$ is plotted against the iteration number k, where f^k is the primal objective value at iteration k. The original, blurry/noisy, and restored (by primal-dual Douglas–Rachford) images are shown in Figure 11.

For all methods, close to optimal fixed primal and dual step sizes (and overrelaxation parameters) were selected by trial and error. Primal and dual step sizes were implemented in the algorithms based on Douglas–Rachford by modifying g and A as in (16). The Chambolle–Pock step sizes s and t were chosen to satisfy $tsL^2 = 1$, where $L = (\|K_{\mathrm{p}}\|^2 + \|D\|^2)^{1/2} \approx \|A\|$. The norm of K_{p} can be computed analytically because $K_{\mathrm{p}}^T K_{\mathrm{p}}$ is diagonalized by the discrete Fourier basis. The norm of D is $\sqrt{\alpha}$ because $D^T D = \alpha I$.

The average elapsed time per iteration was 0.55 seconds for Chambolle–Pock, 0.78 seconds for ADMM, 0.60 seconds for primal Douglas–Rachford, and 0.52 seconds for primal-dual Douglas–Rachford.

7. Primal-dual decomposition. The mixed splitting algorithm applies to problems in many areas beyond image reconstruction. For example, one often encounters optimization problems that are "almost" separable: the functions f and g in (1) are block-separable, and $A = B + C$ with B block-diagonal and C sparse. In this situation, evaluating the resolvents of f and g decomposes into independent subproblems which can be solved in parallel. A linear system with coefficient matrix $I + \lambda B^T B$ is also separable and can be solved by solving independent smaller equations. Systems with coefficient matrix $I + \lambda C^T C$ may be solved efficiently by exploiting sparsity. The mixed splitting algorithm therefore leads to primal-dual decomposition schemes for almost separable problems. This type of structure generalizes angular and dual-angular structure [32] found in problems that are separable except for a small number of coupling constraints ("dual decomposition") or coupling variables ("primal decomposition"); see also [10, 38]. The splitting methods discussed in this paper can therefore be viewed as generalized primal-dual decomposition methods.

To illustrate this with an image deblurring application, we consider a spatially varying blurring model with a piecewise constant kernel. We assume the image can be partitioned into rectangular regions such that the blurring kernel is constant on each region (variations on this basic model are discussed in [36]). If pixels are ordered appropriately (one subimage after another), then the blurring operator can be represented by a matrix $K = K_{\mathrm{bd}} + K_{\mathrm{s}}$, where K_{bd} is block-diagonal and K_{s} is sparse, and moreover each block of K_{bd} represents a spatially invariant convolution (applied to the corresponding subimage) using periodic boundary conditions. Linear systems involving the blocks of K_{bd} can be solved efficiently via the fast Fourier transform. Similarly, a discrete gradient operator can be represented by a sum $D = D_{\mathrm{bd}} + D_{\mathrm{s}}$ of a block-diagonal and a sparse matrix, where each block of D_{bd} represents a spatially invariant convolution (applied to a subimage), using periodic boundary conditions.

We consider again the deblurring model (33), where now the blurring kernel is only assumed to be piecewise constant (and replicate boundary conditions are used for the entire image). The matrix A in (34) can be decomposed as $A = B + C$ with

$$(37) \qquad B = \begin{bmatrix} K_{\mathrm{bd}} \\ D_{\mathrm{bd}} \end{bmatrix}, \qquad C = \begin{bmatrix} K_{\mathrm{s}} \\ D_{\mathrm{s}} \end{bmatrix}.$$

$$(f(x^k) - f^\star)/f^\star$$
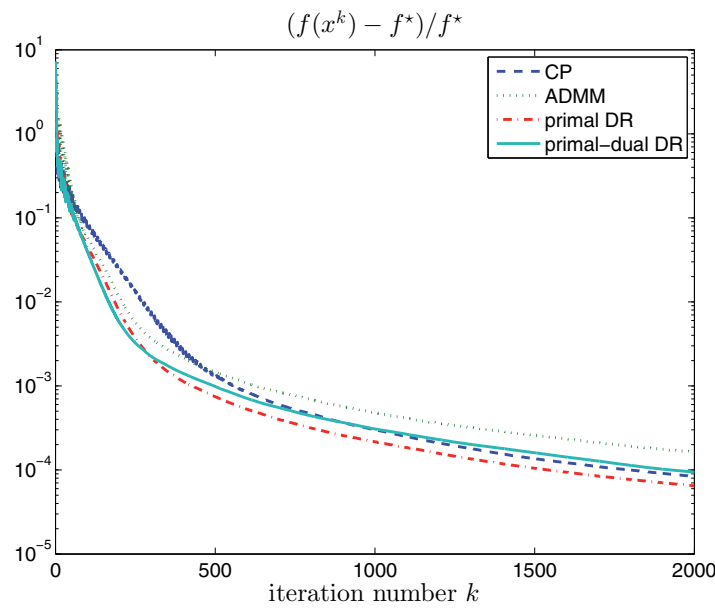
Figure 12. *Relative optimality gap versus iteration number for the experiment in section* 7.

(a) Blurry, noisy image.   (b) After 5 iterations.   (c) Restored image.

Figure 13. *Result for the experiment in section* 7.

Equations with coefficient $I + \lambda B^T B$ (for a given $\lambda > 0$) are easy to solve because $I + \lambda B^T B$ is block-diagonal, with blocks that can be diagonalized by the DFT. Equations with coefficient $I + \lambda C^T C$ can be solved efficiently by exploiting sparsity. We solve (33) using the methods discussed in section 3.3, with the decomposition (37).

Figure 12 compares the performances of the Chambolle–Pock, ADMM, primal Douglas–Rachford, and primal-dual Douglas–Rachford algorithms. The noisy, blurry image b is a degraded version of the 256 by 256 "cameraman" image. The original image (scaled so that intensity values are between 0 and 1) was partitioned into four subimages, each of which was blurred with a 9 by 9 truncated Gaussian kernel. (The standard deviations were $4, 3.5, 3$, and 2 for the upper-left, upper-right,lower-left, and lower-right subimages, respectively.) Then salt and pepper noise was added to a random selection of 10% of the pixels. The parameter γ was

	Primal-dual	Dual
proximal point algorithm	proximal method of multipliers	method of multipliers
Douglas–Rachford splitting	primal-dual Douglas–Rachford splitting	ADMM

chosen to give a visually appealing image reconstruction. A nearly optimal primal objective value $f^\star$ was computed by running the primal-dual Douglas–Rachford algorithm for $10,000$ iterations. In Figure 12 the quantity $(f^k - f^\star)/f^\star$ is plotted against the iteration number k. (f^k is the primal objective value at iteration k.) The blurry/noisy and the restored (by primal-dual Douglas–Rachford) images are shown in Figure 13. The figure also shows the image after five iterations of the primal-dual Douglas–Rachford method. In this partially restored image the artifacts due to the spatially varying blurring operator are still visible.

For all methods, close to optimal fixed primal and dual step sizes (and overrelaxation parameters) were selected by trial and error. Primal and dual step sizes (for the Douglas–Rachford methods) were implemented by modifying g and A as in (16). The Chambolle–Pock step sizes s and t were chosen to satisfy $tsL^2 = 1$, where $L = (\|K_{\mathrm{bd}}\|^2 + 8)^{1/2} \approx \|A\|$. (It can be shown that $\|D\|^2 \le 8$.) The norm of K_{bd} can be computed analytically because the blocks of K_{bd} are diagonalized by the discrete Fourier basis. The average elapsed time per iteration was 0.086 seconds for Chambolle–Pock, 0.15 seconds for ADMM, and 0.12 seconds for both primal Douglas–Rachford and primal-dual Douglas–Rachford.

8. Conclusions. We have presented primal-dual operator splitting algorithms for solving convex optimization problems

$$\text{minimize} \quad f(x) + g(Ax),$$

where f and g are closed, convex functions with inexpensive proximal operators, and the matrix A can be split as a sum $A = B + C$ of two structured matrices. Our approach is to apply the Douglas–Rachford splitting method to the primal-dual optimality conditions (Karush–Kuhn–Tucker conditions), expressed as a monotone inclusion. The relationship of this approach to other well-known methods is illustrated in Table 1. The method of multipliers (or augmented Lagrangian method) [41] and the Bregman iteration [49] can be interpreted as solving the dual optimality condition

$$(38) \qquad\qquad 0 \in \partial g^*(z) - A \partial f^*(-A^T z)$$

using the proximal point algorithm. The ADMM and split Bregman methods can be interpreted as solving the dual optimality condition using the Douglas–Rachford algorithm [20]. The proximal method of multipliers [41] uses the proximal point algorithm to solve the primal-dual optimality conditions. We complete this picture by using the Douglas–Rachford method to solve the primal-dual optimality conditions. This method can therefore be viewed as a primal-dual version of the ADMM or split Bregman methods, or as an alternating direction version of the proximal method of multipliers. The primal-dual approach has the advantage that it lends itself to several interesting splitting strategies. In particular, the "mixed splitting" strategy exploits additive structure $A = B + C$ in the linear term, where B and C have

useful, but different, types of matrix structure. We have illustrated this with applications in image deblurring, in which the matrix B can be diagonalized by a DFT and C is a sparse matrix.

In the numerical experiments we compared the primal-dual Douglas–Rachford splitting method with three other methods: the Douglas–Rachford method applied to a reformulated primal problem (also known as Spingarn's method), the ADMM algorithm (Douglas–Rachford applied to the dual), and the Chambolle–Pock algorithm. All methods depend on three algorithm parameters: a primal step size, a dual step size, and an overrelaxation parameter. Using different primal and dual step sizes can also be interpreted as first scaling the matrix A and g and then using a single step size. We observed that with a careful tuning of the algorithm parameters the four methods performed similarly. In all experiments, the primal-dual Douglas–Rachford approach is consistently one of the fastest methods. Compared with the primal and dual Douglas–Rachford approaches it has the advantage that it avoids the introduction of large numbers of auxiliary variables and constraints. An advantage over the Chambolle–Pock algorithm is that the step size selection does not require estimates or bounds on the norm of A. This is particularly important in large-scale applications where the norm of the linear operators is difficult to estimate (unlike the norms of the convolution operators encountered in image deblurring). As in ADMM and other applications of the Douglas–Rachford method, it is difficult to provide general guidelines about the choice of step sizes. When comparing algorithms we used constant step sizes, tuned by trial and error. More practical adaptive strategies for step size selection in the Douglas–Rachford method have been proposed in [29, 28].

REFERENCES

[1] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York, 2011.

[2] A. Beck and M. Teboulle, *Fast gradient-based algorithm for constrained total variation image denoising and deblurring problems*, IEEE Trans. Image Process., 18 (2009), pp. 2419–2434.

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

[4] H. Brézis, *Opérateurs maximaux monotones et semi-groupes de contractions dans les espaces de Hilbert*, North–Holland Math. Stud. 5, North–Holland, Amsterdam, 1973.

[5] L. M. Briceño-Arias and P. L. Combettes, *A monotone + skew splitting model for composite monotone inclusions in duality*, SIAM J. Optim., 21 (2011), pp. 1230–1250.

[6] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging Vision, 40 (2011), pp. 120–145.

[7] R. H. Chan, M. Tao, and X. Yuan, *Constrained total variation deblurring models and fast algorithms based on alternating direction method of multipliers*, SIAM J. Imaging Sci., 6 (2013), pp. 680–697.

[8] T. F. Chan, G. H. Golub, and P. Mulet, *A nonlinear primal-dual method for total variation-based image restoration*, SIAM J. Sci. Comput., 20 (1999), pp. 1964–1977.

[9] T. F. Chan, A. M. Yip, and F. E. Park, *Simultaneous total variation image inpainting and blind deconvolution*, Internat. J. Imaging Syst. Tech., 15 (2005), pp. 92–102.

[10] M. CHIANG, S. H. LOW, R. CALDERBANK, AND J. C. DOYLE, *Layering as optimization decomposition: A mathematical theory of network architectures*, Proc. IEEE, 95 (2007), pp. 255–312.

[11] P. L. COMBETTES AND J.-C. PESQUET, *A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery*, IEEE J. Selected Topics Signal Process., 1 (2007), pp. 564–574.

[12] P. L. COMBETTES AND J.-C. PESQUET, *Proximal splitting methods in signal processing*, in Fixed-Point Algorithms for Inverse Problems in Science and Engineering, Springer Optim. Appl. 49, Springer, New York, 2011, pp. 185–212.

[13] P. L. COMBETTES AND J.-C. PESQUET, *Primal-dual splitting algorithm for solving inclusions with mixtures of composite, Lipschitzian, and parallel-sum type monotone operators*, Set-Valued Variational Anal., 20 (2012), pp. 307–330.

[14] P. L. COMBETTES AND V. R. WAJS, *Signal recovery by proximal forward-backward splitting*, Multiscale Model. Simul., 4 (2005), pp. 1168–1200.

[15] L. CONDAT, *A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms*, J. Optim. Theory Appl., 158 (2013), pp. 460–479.

[16] J. ECKSTEIN, *Augmented Lagrangian and Alternating Minimization Methods for Convex Optimization: A Tutorial and Some Illustrative Computational Results*, Tech. report RRR 32-2012, Rutgers Center for Operations Research, Rutgers University, New Brunswick, NJ, 2012.

[17] J. ECKSTEIN AND D. BERTSEKAS, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Program., 55 (1992), pp. 293–318.

[18] E. ESSER, X. ZHANG, AND T. CHAN, *A General Framework for a Class of First Order Primal-Dual Algorithms for TV Minimization*, Tech. report CAM 09-67, Department of Mathematics, UCLA, Los Angeles, CA, 2009.

[19] J. E. ESSER, *Primal Dual Algorithms for Convex Models and Applications to Image Restoration, Registration and Nonlocal Inpainting*, Ph.D. thesis, Department of Mathematics, UCLA, Los Angeles, CA, 2010.

[20] D. GABAY, *Applications of the method of multipliers to variational inequalities*, in Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems, Stud. Math. Appl., M. Fortin and R. Glowinski, eds., North–Holland, Amsterdam, 1983, pp. 299–331.

[21] D. GABAY AND B. MERCIER, *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*, Comput. Math. Appl., 2 (1976), pp. 17–40.

[22] R. GLOWINSKI AND A. MARROCCO, *Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires*, Revue Française Automat. Informat. Recherche Opérationnelle RAIRO Anal. Numer., 9 (1975), pp. 41–76.

[23] T. GOLDSTEIN AND S. OSHER, *The split Bregman method for L1-regularized problems*, SIAM J. Imaging Sci., 2 (2009), pp. 323–343.

[24] M. GRANT AND S. BOYD, *Graph implementations for nonsmooth convex programs*, in Recent Advances in Learning and Control (a Tribute to M. Vidyasagar), V. Blondel, S. Boyd, and H. Kimura, eds., Springer, New York, 2008, pp. 95–110.

[25] M. GRANT AND S. BOYD, *CVX: Matlab Software for Disciplined Convex Programming, version 2.0 (beta)*, http://cvxr.com (2012).

[26] P. C. HANSEN, J. G. NAGY, AND D. P. O'LEARY, *Deblurring Images. Matrices, Spectra, and Filtering*, Fundam. Algorithms 3, SIAM, Philadelphia, 2006.

[27] B. HE AND X. YUAN, *Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective*, SIAM J. Imaging Sci., 5 (2012), pp. 119–149.

[28] B. S. HE, L. Z. LIAO, AND S. L. WANG, *Self-adaptive operator splitting methods for monotone variational inequalities*, Numer. Math., 94 (2003), pp. 715–737.

[29] B. S. HE, H. YANG, AND S. L. WANG, *Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities*, J. Optim. Theory Appl., 106 (2000), pp. 337–356.

[30] G. KUTYNIOK AND D. LABATE, *Shearlets: Multiscale Analysis for Multivariate Data*, Springer, New York, 2012.

[31] G. KUTYNIOK, M. SHAHRAM, AND X. ZHUANG, *Shearlab: A Rational Design of a Digital Parabolic Scaling Algorithm*, preprint, http://arxiv.org/abs/1106.1319, 2011.

[32] L. S. LASDON, *Optimization Theory for Large Systems*, Dover, Mineola, NY, 2002. Reprint of the 1970 original.

[33] P. L. LIONS AND B. MERCIER, *Splitting algorithms for the sum of two nonlinear operators*, SIAM J. Numer. Anal., 16 (1979), pp. 964–979.

[34] S. MALLAT, *A Wavelet Tour of Signal Processing*, 2nd ed., Academic Press, New York, 1999.

[35] J. J. MOREAU, *Proximité et dualité dans un espace hilbertien*, Bull. Math. Soc. France, 93 (1965), pp. 273–299.

[36] J. G. NAGY AND D. P. O'LEARY, *Restoring images degraded by spatially variant blur*, SIAM J. Sci. Comput., 19 (1998), pp. 1063–1082.

[37] M. K. NG, R. H. CHAN, AND W.-C. TANG, *A fast algorithm for deblurring models with Neumann boundary conditions*, SIAM J. Sci. Comput., 21 (1999), pp. 851–866.

[38] D. P. PALOMAR AND M. CHIANG, *A tutorial on decomposition methods for network utility maximization*, IEEE J. Selected Areas Commun., 24 (2006), pp. 1439–1451.

[39] N. PARIKH AND S. BOYD, *Proximal algorithms*, Found. Trends Optim., 1 (2013), pp. 123–231.

[40] R. T. ROCKAFELLAR, *Convex Analysis*, 2nd ed., Princeton University Press, Princeton, NJ, 1970.

[41] R. T. ROCKAFELLAR, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. Oper. Res., 1 (1976), pp. 97–116.

[42] R. T. ROCKAFELLAR, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[43] L. RUDIN, S. J. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.

[44] M. ŠOREL, *Removing boundary artifacts for real-time iterated shrinkage deconvolution*, IEEE Trans. Image Process., 21 (2012), p. 2329.

[45] J. E. SPINGARN, *Partial inverse of a monotone operator*, Appl. Math. Optim., 10 (1983), pp. 247–265.

[46] J. E. SPINGARN, *Applications of the method of partial inverses to convex programming: Decomposition*, Math. Program., 32 (1985), pp. 199–223.

[47] P. TSENG, *A modified forward-backward splitting method for maximal monotone mappings*, SIAM J. Control Optim., 38 (2000), pp. 431–446.

[48] C. R. VOGEL, *Computational Methods for Inverse Problems*, Frontiers Appl. Math. 23, SIAM, Philadelphia, 2002.

[49] W. YIN, S. OSHER, D. GOLDFARB, AND J. DARBON, *Bregman iterative algorithms for ℓ_1-minimization with application to compressed sensing*, SIAM J. Imaging Sci., 1 (2008), pp. 143–168.