Linearized Primal-Dual Methods for Linear Inverse Problems with Total Variation Regularization and Finite Element Discretization

WENYI TIAN* XIAOMING YUAN†

September 20, 2016

Abstract. Linear inverse problems with total variation regularization can be reformulated as saddle-point problems; the primal and dual variables of such a saddle-point reformulation can be discretized in piecewise affine and constant finite element spaces, respectively. Thus, the well-developed primal-dual approach (a.k.a. inexact Uzawa method) is conceptually applicable to such a regularized and discretized model. When the primal-dual approach is applied, the resulting subproblems may be highly nontrivial and it is necessary to discuss how to tackle these hard subproblems and thus make the primal-dual approach implementable. In this paper, we suggest to linearize the data-fidelity quadratic term of the hard subproblems so as to obtain easier subproblems. A linearized primal-dual method is thus proposed. Inspired by the fact that the linearized primal-dual method can be explained as an application of the proximal point algorithm, a relaxed version of the linearized primal-dual method, which can often accelerate the convergence numerically with the same order of computation, is also proposed. The global convergence and worst-case convergence rate measured by the iteration complexity are established for the new algorithms. Their efficiency is verified by some numerical results.

Keywords: Linear inverse problem, Numerical optimization, Saddle-point problem, Primal-dual method, Total variation, Finite element, Convergence rate

1 Introduction

Consider the linear equation with discontinuous or piecewise constant solutions

$$Au = g \tag{1.1}$$

with $A : L^r(\Omega) \to L^2(\Omega)$ a bounded linear operator, $\Omega \subset \mathbb{R}^d$ a bounded domain with a Lipschitz continuous boundary, $d = 1, 2$, and $g \in L^2(\Omega)$ a given function. For a variety of practical applications in such areas as astrophysics, signal and image processing, statistical inference, and optics, the problem (1.1) may be ill-posed and thus it cannot be solved directly, see, e.g. [25, 29, 32, 56], for some monographs.

To solve ill-posed cases of (1.1), regularization techniques are important. Among the different regularization techniques in the literature, the total variation (TV) regularization is widely used, particularly for inverse problems with discontinuous solutions. Indeed, since the work [31, 52], TV regularization has found many applications in areas such as image

*Center for Applied Mathematics, Tianjin University, Tianjin 300072, China; Department of Mathematics, Hong Kong Baptist University, Hong Kong, China. Email: twymath@gmail.com

†Department of Mathematics, Hong Kong Baptist University, Hong Kong, China. This author was partially supported by the General Research Fund from Hong Kong Research Grants Council: HKBU 12300515. Email: xmyuan@hkbu.edu.hk

processing [2, 12, 13, 14, 16, 20, 24, 44], linear operator inverse problems [1, 18, 51, 58], parameters identification in partial differential equations [17, 19, 23, 37], and so on. An important property of the TV functional is exhibiting a spatially sparse gradient. Thus it can effectively recover solutions with large constant regions and sharp edges for models whose unknown variables describe the density or material functions changing in different regions or objects. We refer to [12, 13, 24, 51, 53] for some theoretical discussions on the TV regularization; and [14, 17, 19, 24, 52, 53] for some numerical results of its ability of restoring discontinuities near sharp edges.

In this paper, we also consider the TV regularization for the inverse problem (1.1). That is, we approximate a solution of (1.1) by solving the minimization model

$$\inf_u \Big\{ E(u) := \frac{1}{2}\|Au-g\|^2_{L^2(\Omega)} + \alpha\|Du\| \Big\}. \tag{1.2}$$

In (1.2), $\frac{1}{2}\|Au-g\|^2_{L^2(\Omega)}$ is the data-fidelity term reflecting the purpose of solving the linear system (1.1); $\|Du\|$ is the TV regularization term defined by

$$\|Du\| := \sup\Big\{ \int_\Omega u \,\mathrm{div}\varphi \,\mathrm{d}x : \varphi \in C^1_c(\Omega;\mathbb{R}^d), \|\varphi\|_\infty \le 1 \Big\}, \tag{1.3}$$

where $\|\varphi\|_\infty = \sup_{x\in\Omega}(\sum_{i=1}^d |\varphi_i(x)|^2)^{1/2}$, Du represents the gradient of u in the distributional sense, div denotes the divergence operator, $C^1_c(\Omega;\mathbb{R}^d)$ is the set of once continuously differentiable $\mathbb{R}^d$-valued functions with compact support in Ω; and $\alpha > 0$ is a parameter reflecting the relative weight of the data-fidelity and regularization terms. Note that the $BV(\Omega)$ space endowed with norm $\|v\|_{BV} := \|v\|_{L^1(\Omega)} + \|Dv\|$ is a Banach space, see, e.g., [3, 5, 6, 62].

The model (1.2) is not easy to solve, mainly due to the nonsmoothness of the TV regularization term. We follow the approach in [7, 8, 15] to replace the TV term by its dual representation. More specifically, the model (1.2) can be reformulated as the saddle-point problem

$$\inf_u E(u) = \inf_u \sup_p \Big\{ \mathcal{E}(u,p) := \frac{1}{2}\|Au-g\|^2_{L^2(\Omega)} + \alpha\int_\Omega \nabla u\cdot p \,\mathrm{d}x - I_B(p) \Big\}, \tag{1.4}$$

where $B = \{p \in L^1(\Omega;\mathbb{R}^d) : \|p\|_\infty \le 1\}$ and $I_B(\cdot)$ denotes its indicator function.

As mentioned in [7], the piecewise constant and piecewise affine globally continuous finite element spaces are dense in $BV(\Omega)$ with respect to weak* convergence in $BV(\Omega)$; and the space $BV(\Omega)$ is continuously embedded into $L^r(\Omega)$ for all r satisfying $1 \le r \le \frac{d}{d-1}$, see Theorem 10.1.4 in [5]. In addition, it was demonstrated in [7] that the piecewise constant finite element approximation for u cannot be expected to converge to an exact solution in general. Thus, the following finite element spaces

$$\begin{cases} \mathcal{S}^1(\mathcal{T}_h) := \{v_h \in C(\bar\Omega) : v_h|_T \text{ is affine for each } T \in \mathcal{T}_h\}, \\ \mathcal{L}^0(\mathcal{T}_h) := \{q_h \in L^1(\Omega) : q_h|_T \text{ is constant for each } T \in \mathcal{T}_h\}, \end{cases} \tag{1.5}$$

are built to approximate the functions u and p in (1.4), respectively; where $\mathcal{T}_h$ denotes a regular triangulation of Ω into intervals or triangles and $h = \max_{T\in\mathcal{T}_h}\mathrm{diam}(T)$ as the maximal diameter. It is easy to see that

$$\mathcal{L}^0(\mathcal{T}_h)^d := \underbrace{\mathcal{L}^0(\mathcal{T}_h)\times\cdots\times\mathcal{L}^0(\mathcal{T}_h)}_{d}$$

is a space of piecewise constant vector fields equipped with L^2 scalar product as

$$(p_h, q_h) = \sum_{i=1}^{d} \int_\Omega (p_h)_i (q_h)_i \mathrm{d}x,$$

where $(p_h)_i$ stands for the i-th component of vector-valued function p_h. Furthermore, it is shown in [7] that

$$\|Du_h\| = \sup_{p_h \in \mathcal{L}^0(\mathcal{T}_h)^d, \|p_h\|_\infty \le 1} \int_\Omega \nabla u_h \cdot p_h \, \mathrm{d}x.$$

Then, the saddle-point problem (1.4) approximated in the finite element spaces given by (1.5) can be reformulated as the following discretized version:

(1.6)
$$\begin{aligned} \inf_{u_h} E(u_h) &= \inf_{u_h \in \mathcal{S}^1(\mathcal{T}_h)} \sup_{p_h \in \mathcal{L}^0(\mathcal{T}_h)^d} \mathcal{E}(u_h, p_h) \\ &= \inf_{u_h \in \mathcal{S}^1(\mathcal{T}_h)} \sup_{p_h \in \mathcal{L}^0(\mathcal{T}_h)^d} \left\{ \frac{1}{2} \|Au_h - g\|^2_{L^2(\Omega)} + \alpha \int_\Omega \nabla u_h \cdot p_h \, \mathrm{d}x - I_B(p_h) \right\}. \end{aligned}$$

In this paper we focus on the discretized saddle-point problem (1.6) arising from (1.2) and discuss how to solve it efficiently. In particular, we concentrate on the implementation of primal-dual methods, which can be traced back to the inexact Uzawa method in [4], to the discretized saddle-point problem (1.6). Recently, various primal-dual methods have been studied in different contexts, see, e.g., [15, 26, 34, 61] for some special considerations for some variational image restoration models with TV regularization and some more general variants in [10, 50]. Conceptually all these existing primal-dual schemes are applicable to the discretized saddle-point problem (1.6) under our discussion; while as we shall show later, some numerical issues relating to the particular problem (1.6) are worthy of more sophisticated discussions and this consideration motivates us to propose two new primal-dual schemes that are particularly suitable for the discretized saddle-point problem (1.6) in the finite element discretization setting. Let us concentrate our discussion on a generalized primal-dual method proposed in [15] whose iterative scheme for (1.6) reads as

(1.7)
$$\begin{cases} u_h^{n+1} = \displaystyle\arg\min_{u_h \in \mathcal{S}^1(\mathcal{T}_h)} \left\{ \mathcal{E}(u_h, p_h^n) + \frac{1}{2\tau} \|u_h - u_h^n\|^2_{L^2(\Omega)} \right\}, \\ \tilde{u}_h^{n+1} = 2u_h^{n+1} - u_h^n, \\ p_h^{n+1} = \displaystyle\arg\max_{p_h \in \mathcal{L}^0(\mathcal{T}_h)^d} \left\{ \mathcal{E}(\tilde{u}_h^{n+1}, p_h) - \frac{1}{2\varsigma} \|p_h - p_h^n\|^2_{L^2(\Omega)} \right\}, \end{cases}$$

with $\tau > 0$ and $\varsigma > 0$. We consider this particular primal-dual scheme because of its numerical efficiency that has been well verified in the literature and its theoretical simplicity such as its equivalence to the proximal point algorithm (see [42, 43]) as analyzed in [34]. Some other choices for generating $\tilde{u}_h^{n+1}$ have been studied in [15, 34]; but here we only focus on (1.7) with the simplest choice of $\tilde{u}_h^{n+1}$. Note that the parameters $\tau > 0$ and $\varsigma > 0$ can be understood as the step sizes of implementing gradient-based iterative methods for the minimization and maximization subproblems in (1.7), respectively; and they should be appropriately chosen to ensure the convergence as shown in [34]. Since how to tune these two parameters numerically is not the emphasis in this paper, we restrict our discussion to

the case

$$
\text{(1.8)}\qquad \begin{cases} u_h^{n+1} = \underset{u_h\in\mathcal{S}^1(\mathcal{T}_h)}{\arg\min}\big\{\mathcal{E}(u_h,p_h^n)+\dfrac{1}{2\tau}\|u_h-u_h^n\|^2_{L^2(\Omega)}\big\},\\ \tilde{u}_h^{n+1} = 2u_h^{n+1}-u_h^n,\\ p_h^{n+1} = \underset{p_h\in\mathcal{L}^0(\mathcal{T}_h)^d}{\arg\max}\big\{\mathcal{E}(\tilde{u}_h^{n+1},p_h)-\dfrac{\sigma}{2\tau}\|p_h-p_h^n\|^2_{L^2(\Omega)}\big\}, \end{cases}
$$

where the parameters τ and ς in (1.7) are related by the fixed ratio $\sigma > 0$. This special consideration will alleviate the notation of analysis and thus expose our main content more clearly.

For the special case of (1.6) where A is the identity operator, the scheme (1.8) has been well studied in the literature, see, e.g., [7, 54]. For some cases of (1.1), however, the particular application of primal-dual scheme (1.8) may not be truly implementable. To see this, first of all, we know that the optimality condition of the u_h-subproblem in (1.8) can be written as

$$
\text{(1.9)}\qquad \big(\frac{1}{\tau}(u_h^{n+1}-u_h^n)+A^*(Au_h^{n+1}-g),v_h\big)+\alpha\big(p_h^n,\nabla v_h\big)=0,\ \forall\, v_h\in\mathcal{S}^1(\mathcal{T}_h),
$$

where A^* denotes the adjoint operator of A. With the choice of $\mathcal{S}^1(\mathcal{T}_h)$ given in (1.5), (1.9) reduces to a system of linear equations; and conceptually its solution can be given explicitly by algebra. However, from numerical point of view, it could be hard to find a solution of (1.9) for some important cases of (1.1) and this is indeed our main motivation of this paper. These cases include the one where the operator A is implicit and thus the discretized matrix of the operator A^*A is unknown. Moreover, even if A is known explicitly, it is well known that the dimension of the discretized matrix of A^*A could be easily huge when the function u is discretized over a finite element mesh with a small diameter for achieving high accuracy, see, e.g., [11]. Also, the discretized matrix of A^*A is generally full and it has no special structures [38] and no ad hoc faster solvers for (1.9) can be expected. Thus, solving (1.9) directly may not be practical. All these difficulties urge us to consider how to solve the u_h-subproblem in (1.8) approximately and efficiently; while the convergence can be still ensured.

Our strategy is to replace the regularization term $\frac{1}{2\tau}\|u_h-u_h^n\|^2_{L^2(\Omega)}$ of the u_h-subproblem in (1.8) by the more general term with a metric distance

$$
\frac{1}{2\tau}\|u_h-u_h^n\|^2_{X,L^2(\Omega)},
$$

with $\|\cdot\|^2_{X,L^2(\Omega)} := (X\cdot,\cdot)$, where X is a bounded and strictly monotone operator. That is, instead of (1.8), we suggest the following primal-dual approach:

$$
\text{(1.10)}\qquad \begin{cases} u_h^{n+1} = \underset{u_h\in\mathcal{S}^1(\mathcal{T}_h)}{\arg\min}\big\{\mathcal{E}(u_h,p_h^n)+\dfrac{1}{2\tau}\|u_h-u_h^n\|^2_{X,L^2(\Omega)}\big\},\\ \tilde{u}_h^{n+1} = 2u_h^{n+1}-u_h^n,\\ p_h^{n+1} = \underset{p_h\in\mathcal{L}^0(\mathcal{T}_h)^d}{\arg\max}\big\{\mathcal{E}(\tilde{u}_h^{n+1},p_h)-\dfrac{\sigma}{2\tau}\|p_h-p_h^n\|^2_{L^2(\Omega)}\big\}. \end{cases}
$$

An interesting choice for X is $X := I-\tau A^*A$, where I is the identity operator. With this choice, the u_h-subproblem in (1.10) is specified as

$$
\text{(1.11)}\quad u_h^{n+1} = \underset{u_h\in\mathcal{S}^1(\mathcal{T}_h)}{\arg\min}\Big\{\big(A^*(Au_h^n-g),u_h\big)+\alpha\int_\Omega \nabla u_h\cdot p_h^n\,\mathrm{d}x+\frac{1}{2\tau}\|u_h-u_h^n\|^2_{L^2(\Omega)}\Big\},
$$

Thus, the u_h-subproblem in (1.8) is approximated by (1.11), meaning the data-fidelity term $\frac{1}{2}\|Au_h - g\|^2_{L^2(\Omega)}$ in (1.8) is linearized. Note that the linearized subproblem (1.11) is equivalent to the system of equations

$$\Big(\frac{1}{\tau}(u_h^{n+1} - u_h^n) + A^*(Au_h^n - g), v_h\Big) + \alpha\big(p_h^n, \nabla v_h\big) = 0, \ \forall\, v_h \in \mathcal{S}^1(\mathcal{T}_h). \tag{1.12}$$

For some cases such as the inverse source or optimal control problem of elliptic partial differential equations (see e.g., [21, 38]), the operator A is implicit but $A^*(Au_h^n - g)$ can be computed by its dual equation, see, e.g., [41, 57]. In this case, (1.12) can be solved easily. Even when the operator A is explicit, solving (1.12) does not require computing the inverse of any matrix involving the discretized matrix of A^*A. Therefore, compared with the counterpart in (1.8), the u_h-subproblem in (1.10) with the particular choice $X := I - \tau A^*A$, i.e., (1.12), could be much easier.

The rest of this paper is organized as follows. In Section 2, we revisit the linearized primal-dual method (1.10) from the perspective of proximal point algorithm (PPA); and propose a relaxed version of (1.10), followed by some remarks. In Section 3, we consider the finite element approximation for the model (1.2), and the error of the finite element approximation to the energy functional $E(\cdot)$ is estimated. Then, we prove the convergence and estimate the convergence rate measured by the iteration complexity for the new methods in Section 4. Some preliminary numerical results are reported in Section 5 to verify the efficiency of the proposed methods. Finally, some conclusions are made in Section 6.

2 Algorithms

In this section, we propose two primal-dual-based algorithms for the discretized saddle-point problem (1.6) and give some remarks. We first propose a new linearized primal-dual method and then follow the work [34] to show that it is indeed an application of the PPA in [42, 43]. Then, we follow the relaxed version of PPA in [28] to propose a relaxed version of the linearized primal-dual method.

2.1 A Linearized Primal-Dual Method

We first explicitly present the linearized primal-dual scheme in which the u_h-subproblem in (1.8) is approximated by (1.11). It is summarized in Algorithm 1.

Remark 2.1. *The u_h-subproblem (2.2a) is equivalent to (1.12); and it can be easily solved as mentioned previously. For the p_h-subproblem (2.2c), as mentioned in [7], its solution is explicitly given by*

$$p_h^{n+1} = \big(p_h^n + (\alpha\tau/\sigma)\nabla\tilde{u}_h^{n+1}\big)/\max\big\{1, |p_h^n + (\alpha\tau/\sigma)\nabla\tilde{u}_h^{n+1}|\big\}$$

in component-wise.

2.2 The PPA Revisit

Now, we follow the work [34] to show that Algorithm 1 can be regarded as an application of the PPA in [42, 43]. This analysis requires the variational form of the subproblems in Algorithm 1. First, let us analyze the optimality condition of the saddle-point problem (1.6) as a variational inequality in the following lemma. Hereafter, the notation $(\cdot,\cdot)$ stands for the L^2 scalar product.

Algorithm 1: A Linearized primal-dual scheme for solving (1.6).

Input: Choose an initial iteration $(u_h^0, p_h^0) \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{L}^0(\mathcal{T}_h)^d$. Choose $\tau > 0$ and $\sigma > 0$ such that

$$\Big(\frac{1}{\tau} - \|A\|^2\Big)\frac{\sigma}{\tau} > \alpha^2\|\nabla\|^2, \tag{2.1}$$

for $n = 0, 1, 2, \cdots,$ **do**

Generate the new iteration (u_h^{n+1}, p_h^{n+1}) via solving

$$u_h^{n+1} = \operatorname*{arg\,min}_{u_h \in \mathcal{S}^1(\mathcal{T}_h)} \left\{ \begin{array}{c} \big(A^*(Au_h^n - g), u_h\big) + \alpha \displaystyle\int_\Omega \nabla u_h \cdot p_h^n \,\mathrm{d}x \\ + \frac{1}{2\tau}\|u_h - u_h^n\|^2_{L^2(\Omega)} \end{array} \right\}, \tag{2.2a}$$

$$\tilde{u}_h^{n+1} = 2u_h^{n+1} - u_h^n, \tag{2.2b}$$

$$p_h^{n+1} = \operatorname*{arg\,max}_{p_h \in \mathcal{L}^0(\mathcal{T}_h)^d} \Big\{ \alpha \int_\Omega \nabla \tilde{u}_h^{n+1} \cdot p_h \,\mathrm{d}x - I_B(p_h) - \frac{\sigma}{2\tau}\|p_h - p_h^n\|^2_{L^2(\Omega)} \Big\}. \tag{2.2c}$$

end

Lemma 2.1. *The function $u_h \in \mathcal{S}^1(\mathcal{T}_h)$ minimizes $E(\cdot)$ in $\mathcal{S}^1(\mathcal{T}_h)$ if and only if there exists $p_h \in \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d) := \{q_h \in \mathcal{L}^0(\mathcal{T}_h)^d : \|q_h\|_\infty \le 1\}$ such that*

$$\begin{aligned} \big(A^*(Au_h - g), v_h\big) + \alpha\big(p_h, \nabla v_h\big) = 0, &\quad \forall\, v_h \in \mathcal{S}^1(\mathcal{T}_h), \\ \big(\nabla u_h, q_h - p_h\big) \le 0, &\quad \forall\, q_h \in \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d). \end{aligned} \tag{2.3}$$

Proof. The equations in (2.3) are the Karush-Kuhn-Tucker optimality conditions of $\mathcal{E}(\cdot,\cdot)$. The proof is similar as that of Lemma 10.3 in [8]; thus omitted. □

With Lemma 2.1, we can rewrite (2.3) in a compact form: Finding $\mu_h \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d)$ such that

$$\big(F(\mu_h), \nu_h - \mu_h\big) \ge 0, \quad \forall\, \nu_h \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d), \tag{2.4}$$

where

$$\mu_h = \begin{pmatrix} u_h \\ p_h \end{pmatrix}, \quad \nu_h = \begin{pmatrix} v_h \\ q_h \end{pmatrix}, \quad F(\mu_h) = \begin{pmatrix} -\alpha \mathrm{div} p_h + A^*(Au_h - g) \\ -\alpha \nabla u_h \end{pmatrix}, \tag{2.5}$$

and $-\mathrm{div}$ is the adjoint operator of ∇ and $-(\mathrm{div} q_h, v_h) = (q_h, \nabla v_h)$. It is easy to check that the mapping $F(\cdot)$ in (2.5) satisfies

$$\big(F(\mu_h) - F(\nu_h), \mu_h - \nu_h\big) = \|A(u_h - v_h)\|^2_{L^2(\Omega)}. \tag{2.6}$$

To see why Algorithm 1 is indeed an application of the PPA, we first derive the first-order optimality conditions for the subproblems (2.2a) and (2.2c). It is easy to see that the iterate pair (u_h^{n+1}, p_h^{n+1}) generated by Algorithm 1 satisfies the following conditions:

$$\big(\frac{1}{\tau}(u_h^{n+1} - u_h^n) + A^*(Au_h^n - g), v_h\big) + \alpha\big(p_h^n, \nabla v_h\big) = 0, \ \ \forall\, v_h \in \mathcal{S}^1(\mathcal{T}_h), \tag{2.7}$$

$$\big(-\frac{\sigma}{\tau}(p_h^{n+1} - p_h^n) + \alpha \nabla \tilde{u}_h^{n+1}, q_h - p_h^{n+1}\big) \le 0, \ \ \forall\, q_h \in \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d), \tag{2.8}$$

which can be further written as

$$\left(F(\mu_h^{n+1}) + M(\mu_h^{n+1} - \mu_h^n), \nu_h - \mu_h^{n+1}\right) \geq 0, \quad \forall\, \nu_h \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d) \tag{2.9}$$

with

$$M = \begin{pmatrix} \frac{1}{\tau} I - A^* A & \alpha \mathrm{div} \\ -\alpha \nabla & \frac{\sigma}{\tau} I \end{pmatrix}. \tag{2.10}$$

Therefore, the iteration of Algorithm 1 can be expressed by the variational inequality (2.9). Recall the variational inequality reformulation (2.4) of the saddle-point problem (1.6). It is thus clear that Algorithm 1 can be regarded as an application of the PPA to the variational inequality (2.4) with the metric proximal term defined by the matrix form operator M in (2.10), provided that the parameters of M are chosen to ensure its symmetry and positive definiteness. Indeed, this explains why restricts the choices of τ and σ according to condition (2.1). We refer to [30, 34] for more details.

Finally, we remark that some preconditioned versions of primal-dual schemes (e.g., [50]) can also be analytically explained as applications of the PPA with certain metric proximal terms via the variational inequality context. But the proximal matrices of these primal-dual schemes are mainly chosen to precondition the related linear systems, and they are different from the matrix form operator M in (2.10) whose exclusive purpose is to linearize the data-fidelity term in the objective function so as to alleviate the hard subproblem (1.9) as the easier one (1.12). On the other hand, despite different choices of the proximal matrices for distinct purposes, the unified PPA illustration via the variational inequality context makes the convergence analysis of some primal-dual schemes in different settings, including the preconditioned version in [50] and Algorithm 1 in the finite element discretization setting, significantly easier.

2.3 A Relaxed Linearized Primal-Dual Method

An advantage of explaining Algorithm 1 as an application of the PPA is that we can immediately use the relaxed PPA originally proposed in [28] to propose a relaxed linearized primal-dual method. We summarize the specific steps in Algorithm 2.

Algorithm 2: A relaxed linearized primal-dual scheme for solving (1.6).

Input: Choose an initial iteration $(u_h^0, p_h^0) \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{L}^0(\mathcal{T}_h)^d$. Choose $\tau > 0$ and $\sigma > 0$ by the condition (2.1). Choose $\rho \in (0, 2)$.

for $n = 0, 1, 2, \cdots$, **do**

Primal-dual step Implement the primal-dual scheme (2.2) and denote its output by $\bar{\mu}_h^n = (\bar{u}_h^n; \bar{p}_h^n)$.

Relaxation step Generate the new iterate μ_h^{n+1} by

$$\mu_h^{n+1} = (1 - \rho)\mu_h^n + \rho \bar{\mu}_h^n. \tag{2.11}$$

end

Remark 2.2. *Algorithm 2 differs from Algorithm 1 slightly in the relaxation step* (2.11) *whose extra computation is negligible. As verified empirically in the literature (see, e.g., [34]), this relaxation step can accelerate the convergence favorably. We are thus interested in this relaxed linearized primal-dual method. Its acceleration effectiveness over Algorithm 1 will be further verified by some preliminary numerical results.*

2.4 Comparison with Algorithms in [54]

First, notice that the model discussed in our previous work [54] is a special case of the model (1.2) with $A = I$. Second, though the algorithms proposed in [54] are also inspired by primal-dual methods, they are different from Algorithms 1 and 2 in this paper. To elaborate on the difference, we notice that Algorithm 1 in [54] extends the primal-dual scheme proposed in [15] to the setting of (1.6) with $A = I$ and its iterative scheme is

$$\left\{\begin{aligned} u_h^{n+1} &= \underset{u_h\in\mathcal{S}^1(\mathcal{T}_h)}{\arg\min}\Big\{\mathcal{E}_I(u_h,p_h^n)+\frac{1}{2\tau}\|u_h-u_h^n\|^2_{L^2(\Omega)}\Big\},\\ \tilde{u}_h^{n+1} &= u_h^{n+1}+\theta(u_h^{n+1}-u_h^n),\\ p_h^{n+1} &= \underset{p_h\in\mathcal{L}^0(\mathcal{T}_h)^d}{\arg\max}\Big\{\mathcal{E}_I(\tilde{u}_h^{n+1},p_h)-\frac{\sigma}{2\tau}\|p_h-p_h^n\|^2_{L^2(\Omega)}\Big\}, \end{aligned}\right. \tag{2.12}$$

with

$$\mathcal{E}_I(u_h,p_h) := \frac{\alpha}{2}\|u_h-g\|^2_{L^2(\Omega)}+\int_\Omega \nabla u_h\cdot p_h\,\mathrm{d}x - I_B(p_h)$$

and $\theta\in[-1,1]$ as a combination parameter. Hence, Algorithm 1 in this paper differs from (2.12) in that the combination parameter $\theta = 1$ in (2.2b) rather than a general value in $[-1,1]$ as in (2.12). Another more significant is that, as elucidated in the introduction, the regularization term $\frac{1}{2\tau}\|u_h-u_h^n\|^2_{L^2(\Omega)}$ of the u_h-subproblem in (1.8) is replaced by the more general term with a metric distance

$$\frac{1}{2\tau}\|u_h-u_h^n\|^2_{I-\tau A^*A,L^2(\Omega)}.$$

Hence, the data-fidelity term $\frac{1}{2}\|Au_h-g\|^2_{L^2(\Omega)}$ in (1.8) is linearized in Algorithm 1 in this paper and the resulting subproblem is (2.2a). As comparison, there is no need to change the data-fidelity term for (2.12) in its u_h-subproblem because $A = I$ in this case. To summarize, Algorithm 1 in this paper only considers the combination parameter $\theta = 1$ but for a more general model; while (2.12) considers the more general case of $\theta\in[-1,1]$ but for a specific model; the former considers a new (approximated) data-fidelity term (because $A\neq I$) while the latter keeps the original data-fidelity term (because $A = I$).

Moreover, as analyzed in Section 2.2, Algorithm 1 in this paper can be regarded as an application of the PPA in [42, 43]; and this is not true when $\theta\neq 1$ as shown in [34]. Hence this fact inspires us to apply the relaxed PPA in [28] to propose Algorithm 2 in this paper. As comparison, Algorithm 2 in [54] is in different nature and proposed by different motivations. Let us explain this difference with more details. For (2.12), we show in [54] that the step size τ and the finite element mesh size h should satisfy the condition $\tau\leq ch^2$ for $\theta\in[-1,1)$ where c is a constant, which seems restrictive in the sense of Remark 3.2. Then, with the purpose of relaxing the condition $\tau\leq ch^2$ to $\tau\leq ch$ for $\theta\in[-1,1)$, we propose Algorithm 2 in [54] as a prediction-correction framework which uses the output of (2.12) as a predictor and then corrects it with a particularly designed correction step. Therefore, Algorithm 2 in this paper is in PPA nature and can be regarded as an extension of the work [28] to the setting of (1.6) with a general A; while Algorithm 2 in [54] is in prediction-correction nature with the particular purpose of relaxing the restriction on the step size τ and finite element mesh size h.

Overall, even though both the current work and the previous one [54] are inspired by primal-dual methods, neither of these works is a special case of the other. The algorithms in these two works have significant differences and these differences require us to conduct

the convergence analysis using different techniques and procedures. In these two works, we do purposively use some identical notation referring to variables that play the same roles in their respectively different analysis so that their limited similarity can be exposed more clearly.

3 Finite Element Error Analysis

In this section, we first recall some known results about the existence and uniqueness of the solution to the model (1.2). We then consider the finite element approximation for it, and finally estimate the error of the finite element approximation to the energy functional $E(\cdot)$.

First of all, the existence and uniqueness of the solution point of the regularized model for (1.1) have been studied in the literature, see, e.g., [1, 18, 58]. We summarize the results in the following theorem without proof.

Theorem 3.1. *Let K be a closed and convex subset of $L^r(\Omega)$ with $1 \leq r < d/(d-1)$ for $d \geq 2$ and $1 \leq r < +\infty$ for $d = 1$, then there exists a solution to (1.2) in K. If moreover, the null space $Ker(A)$ of A does not contain nonzero constant functions over Ω, then the functional in (1.2) has a unique minimizer over K.*

Next we will discuss the error estimate of the finite element approximation to the energy functional $E(\cdot)$ used in the primal-dual scheme (1.10) for the two-dimensional case. A definition is recalled at the beginning, see, e.g., [22].

Definition 3.1. *The Lipschitz space $\mathrm{Lip}(\beta, L^2(\Omega))$ with $0 < \beta \leq 1$ consists of all functions $v \in L^2(\Omega)$ such that*

$$|v|_{\mathrm{Lip}(\beta,L^2(\Omega))} := \sup_{t>0}\{t^{-\beta}\omega(v,t)\} < \infty,$$

where $\omega(v,t) = \sup_{|y|\leq t} \left(\int_\Omega |v(x+y) - v(x)|^2 \mathrm{d}x \right)^{1/2}$ is called the first order modulus of smoothness of $v \in L^2(\Omega)$.

For the following theorem, the special case of $A = I$ has been proved in [7] and we can easily follow the proof therein to prove the general case of $A \neq I$. Here, we provide an easier proof which is mainly based on the result (3.3).

Theorem 3.2. *Assume $\Omega \subset \mathbb{R}^2$. Let $u \in L^2(\Omega) \cap BV(\Omega)$ and $u_h \in \mathcal{S}^1(\mathcal{T}_h)$ be the minimizer of the energy functional $E(\cdot)$ in (1.2). If $u \in \mathrm{Lip}(\beta, L^2(\Omega))$ for some $0 < \beta \leq 1$, then we have*

$$E(u_h) - E(u) \leq c h^{\beta/(\beta+1)}. \tag{3.1}$$

Proof. Denote u_ε as the mollification of u and $\mathcal{I}_h u_\varepsilon \in \mathcal{S}^1(\mathcal{T}_h)$ as the nodal interpolation of u_ε. Then, it follows from [59] that

$$\begin{aligned} \|u_\varepsilon - u\|_{L^2(\Omega)} &\leq c|u|_{\mathrm{Lip}(\beta,L^2(\Omega))}\varepsilon^\beta, \\ \|\mathcal{I}_h u_\varepsilon - u_\varepsilon\|_{L^2(\Omega)} &\leq c|u|_{\mathrm{Lip}(\beta,L^2(\Omega))}h^\beta. \end{aligned}$$

We thus have the estimate

$$\begin{aligned} \|\mathcal{I}_h u_\varepsilon - u\|_{L^2(\Omega)} &\leq \|\mathcal{I}_h u_\varepsilon - u_\varepsilon\|_{L^2(\Omega)} + \|u_\varepsilon - u\|_{L^2(\Omega)} \\ &\leq c(h^\beta + \varepsilon^\beta). \end{aligned} \tag{3.2}$$

Further, it follows from Lemma 10.1 in [8] that

$$\|\nabla \mathcal{I}_h u_\varepsilon\|_{L^1(\Omega)} \leq (1 + ch/\varepsilon + c\varepsilon)\|Du\|. \tag{3.3}$$

Since u_h is the minimizer of the energy functional $E(\cdot)$ in $\mathcal{S}^1(\mathcal{T}_h)$, the estimates (3.2) and (3.3) imply that

(3.4)

$$\begin{aligned} E(u_h) - E(u) &\leq E(\mathcal{I}_h u_\varepsilon) - E(u) \\ &= \alpha\|\nabla \mathcal{I}_h u_\varepsilon\|_{L^1(\Omega)} + \frac{1}{2}\|A(\mathcal{I}_h u_\varepsilon) - g\|^2_{L^2(\Omega)} - \alpha\|Du\| - \frac{1}{2}\|Au - g\|^2_{L^2(\Omega)} \\ &= \alpha\|\nabla \mathcal{I}_h u_\varepsilon\|_{L^1(\Omega)} - \alpha\|Du\| + \frac{1}{2}\big(A(\mathcal{I}_h u_\varepsilon + u) - 2g, A(\mathcal{I}_h u_\varepsilon - u)\big) \\ &\leq c(h\varepsilon^{-1} + \varepsilon)\|Du\| + \frac{c}{2}\|A(\mathcal{I}_h u_\varepsilon + u) - 2g\|_{L^2(\Omega)}\|\mathcal{I}_h u_\varepsilon - u\|_{L^2(\Omega)} \\ &\leq c(h\varepsilon^{-1} + \varepsilon + h^\beta + \varepsilon^\beta), \end{aligned}$$

where the last inequality follows from the fact that the operator A, $\|Du\|$ and $\|\mathcal{I}_h u_\varepsilon\|_{L^2(\Omega)}$ are bounded provided that $h \leq c\varepsilon$. Setting $\varepsilon = h^{1/(\beta+1)}$, the proof is complete. □

Remark 3.1. *Let u and u_h be the solution points of (1.2) in the spaces $L^2(\Omega) \cap BV(\Omega)$ and $\mathcal{S}^1(\mathcal{T}_h)$, respectively. When A in (1.2) is the identity operator and $E(\cdot)$ is strongly convex, it is derived in [7, Theorem 3.1] that*

$$\|u - u_h\|^2_{L^2(\Omega)} \leq ch^{\beta/(1+\beta)}.$$

For the general case where the operator A in (1.2) is not identity, we need more assumptions to estimate the error between u and u_h. For example, if A^*A is further assumed to be strongly monotone:

$$\|Aw - Av\|^2 \geq c_0\|w - v\|^2, \quad \forall\, w, v, \tag{3.5}$$

with $c_0 > 0$, as u is a minimizer of $E(\cdot)$, then it follows from the strong convexity of $\|\cdot\|^2_{L^2(\Omega)}$, the convexity of the TV norm and (3.5) that

$$\frac{c_0}{2}\|u_h - u\|^2_{L^2(\Omega)} \leq \frac{1}{2}\|Au_h - Au\|^2_{L^2(\Omega)} \leq E(u_h) - E(u).$$

Together with (3.1), these inequalities imply that

$$\|u - u_h\|^2_{L^2(\Omega)} \leq \frac{2}{c_0}\big(E(u_h) - E(u)\big) \leq \frac{2c}{c_0}h^{\beta/(1+\beta)}.$$

Remark 3.2. *In [49], the PPA is interpreted as a discretization of the gradient flow related to some monotone operator. We can specify this interpretation for the saddle-point problem (1.4) and Algorithm 1. Indeed, as mentioned, the optimality conditions of the subproblems (2.2a) and (2.2c) are given in (2.7) and (2.8), respectively. So, they can be viewed as the discretization of the simultaneous gradient flow of (1.4)*

$$\partial_t u - \alpha \mathrm{div} p + A^*(Au - g) = 0 \tag{3.6}$$

and

$$-\sigma\partial_t p + \alpha\nabla u \in \partial I_B(p), \tag{3.7}$$

respectively, with finite element discretization in space, where ∂_t denotes the time partial derivative and $\partial I_B(p)$ denotes the subdifferential of $I_B(p)$. Moreover, the parameter τ behaves as the step size of the discretization in time direction.

4 Convergence Analysis

In this section, we prove the convergence for the proposed algorithms and estimate their worst-case convergence rates measured by the iteration complexity. Since Algorithm 1 is a special case of Algorithm 2 with $\rho = 1$, it suffices to conduct the analysis only for Algorithm 2.

4.1 Global Convergence

We have shown that Algorithm 1 is an application of the PPA in [42, 43] and Algorithm 2 is obtained directly by using the technique in [28]; thus their convergence can be basically derived from the known results in these literatures. But because we are considering the particular finite element discretization setting (1.6) and it seems worthy to explicitly show the proof in detail; we still include a complete proof of the convergence.

First, as mentioned, the output of the primal-dual step $\bar{\mu}_h^n = (\bar{u}_h^n; \bar{p}_h^n)$ satisfies the variational inequality

$$\big(F(\bar{\mu}_h^n) + M(\bar{\mu}_h^n - \mu_h^n), \nu_h - \bar{\mu}_h^n\big) \geq 0, \quad \forall\, \nu_h \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d), \tag{4.1}$$

where M is given in (2.10). In the next theorem, we prove that the sequence $\{\mu_h^n\}$ generated by Algorithm 2 is strictly contractive with respect to the solution set of (1.6). This is an important proposition to ensure the global convergence of Algorithm 2.

Theorem 4.1 (Strict contraction)**.** *Let μ_h be a solution point of (1.6) and $\{\mu_h^{n+1}\}$ be the sequence generated by Algorithm 2 with $\rho \in (0, 2)$. Then, we have*

$$\|\mu_h^{n+1} - \mu_h\|_M^2 \leq \|\mu_h^n - \mu_h\|_M^2 - \rho(2-\rho)\|\mu_h^n - \bar{\mu}_h^n\|_M^2, \tag{4.2}$$

where M is defined in (2.10).

Proof. Adding $\big(F(\nu_h), \bar{\mu}_h^n - \nu_h\big)$ to both sides of (4.1), we have

$$\begin{aligned}\big(F(\bar{\mu}_h^n) - F(\nu_h) + M(\bar{\mu}_h^n - \mu_h^n), \nu_h - \bar{\mu}_h^n\big) \geq \big(F(\nu_h), \bar{\mu}_h^n - \nu_h\big),\\ \forall\, \nu_h \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d).\end{aligned}$$

Then, using (2.6), we have

$$\begin{aligned}\big(M(\mu_h^n - \bar{\mu}_h^n), \mu_h^n - \nu_h\big) \geq \big(M(\mu_h^n - \bar{\mu}_h^n), \mu_h^n - \bar{\mu}_h^n\big) + \big(F(\nu_h), \bar{\mu}_h^n - \nu_h\big)\\ + \|A(v_h - \bar{u}_h^n)\|_{L^2(\Omega)}^2,\ \forall\, \nu_h \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d),\end{aligned}$$

which implies that

$$\begin{aligned}&2\rho\big(M(\mu_h^n - \bar{\mu}_h^n), \mu_h^n - \nu_h\big) - \rho^2\|\mu_h^n - \bar{\mu}_h^n\|_M^2\\ &\geq \rho(2-\rho)\|\mu_h^n - \bar{\mu}_h^n\|_M^2 + 2\rho\big(F(\nu_h), \bar{\mu}_h^n - \nu_h\big),\\ &\forall\, \nu_h \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d).\end{aligned} \tag{4.3}$$

With (4.3) and a simple calculation, we have that

$$\begin{aligned}\|\mu_h^{n+1} - \nu_h\|_M^2 &= \|(\mu_h^n - \nu_h) - \rho(\mu_h^n - \bar{\mu}_h^n)\|_M^2\\ &= \|\mu_h^n - \nu_h\|_M^2 - 2\rho\big(M(\mu_h^n - \bar{\mu}_h^n), \mu_h^n - \nu_h\big) + \rho^2\|\mu_h^n - \bar{\mu}_h^n\|_M^2\\ &\leq \|\mu_h^n - \nu_h\|_M^2 - \rho(2-\rho)\|\mu_h^n - \bar{\mu}_h^n\|_M^2 - 2\rho\big(F(\nu_h), \bar{\mu}_h^n - \nu_h\big),\\ &\forall\, \nu_h \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d).\end{aligned} \tag{4.4}$$

Therefore, the result is obtained from (4.4) and (2.4) with $\nu_h = \mu_h$. □

The assertion (4.2) implies that the sequence $\{\mu_h^{n+1}\}$ generated by Algorithm 2 is strictly contractive with respect to the solution set of (1.6), which essentially implies the convergence of the sequence $\{\mu_h^{n+1}\}$. We present the convergence result in the following theorem.

Theorem 4.2 (Global convergence)**.** *Let $\{\mu_h^{n+1} = (u_h^{n+1}; p_h^{n+1})\}$ be the sequence generated by Algorithm 2 with $\rho \in (0, 2)$. Then, the sequence $\{u_h^{n+1}\}$ converges to a minimizer of the problem (1.2) in $\mathcal{S}^1(\mathcal{T}_h)$. If $Ker(A) = \{0\}$, then it converges to the unique minimizer of (1.2) in $\mathcal{S}^1(\mathcal{T}_h)$.*

Proof. For any integer $N > 0$, because of the strict contraction property of the sequence $\{\mu_h^n\}$ shown in (4.2), we have

$$\sum_{n=0}^{N} \|\mu_h^n - \bar{\mu}_h^n\|_M^2 \leq \frac{1}{\rho(2-\rho)} \|\mu_h - \mu_h^0\|_M^2,$$

which yields that

$$\lim_{n\to\infty} \|\mu_h^n - \bar{\mu}_h^n\|_M^2 = 0.$$

Recall that M is positive definite under the condition (2.1). Together with (2.11), we have

$$\lim_{n\to\infty} (\mu_h^{n+1} - \mu_h^n) = \rho \lim_{n\to\infty} (\bar{\mu}_h^n - \mu_h^n) = 0$$

and

$$\lim_{n\to\infty} \mu_h^{n+1} = \lim_{n\to\infty} \bar{\mu}_h^n.$$

Since (4.2), the sequence $\{\mu_h^n\}$ is bounded. Thus, it has a cluster point, e.g., denoted as μ_h^*. Then, substituting it into (4.1), we obtain that

$$\big(F(\mu_h^*), \nu_h - \mu_h^*\big) \geq 0, \quad \forall\, \nu_h \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d),$$

which means μ_h^* is a solution point of (2.4). In addition, if $Ker(A) = \{0\}$, then it follows from (2.6) that μ_h^* is the unique solution of (2.4). The assertion follows immediately from Lemma 2.1. □

4.2 Convergence Rate

In this subsection, we estimate the worst-case $O(\frac{1}{N})$ convergence rate measured by the iteration complexity for the proposed algorithms, where N denotes the iteration counter. We follow [45, 47] and many others, to call a worst-case $O(\frac{1}{N})$ convergence rate by meaning that the accuracy to a solution under certain criteria is of the order $O(\frac{1}{N})$ after N iterations of an iterative scheme; or equivalently, it requires at most $O(\frac{1}{\epsilon})$ iterations to achieve an approximate solution with an accuracy of ϵ. Again, we only present the results for Algorithm 2 because Algorithm 1 is its special case with $\rho = 1$.

4.2.1 Convergence Rate in the Ergodic Sense

First, we introduce a criterion to measure the accuracy of an approximate solution point of the variational inequality (2.4).

Theorem 4.3. *The solution set of variational inequality (2.4) is convex and can be characterized as*

$$\Theta := \bigcap_{\nu_h} \big\{\tilde{\mu}_h \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d) : \big(F(\nu_h), \nu_h - \tilde{\mu}_h\big) \geq 0\big\}. \tag{4.5}$$

Proof. The proof is similar to that of Theorem 2.3.5 in [27] or Theorem 2.1 in [35]. □

According to the criterion in [46] and Theorem 4.3, we say that $\tilde{\mu}_h \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d)$ is an approximate solution of variational inequality (2.4) with an accuracy of ϵ if

$$\big(F(\nu_h), \tilde{\mu}_h - \nu_h\big) \le \epsilon, \quad \forall\, \nu_h \in \mathcal{D}(\tilde{\mu}_h), \tag{4.6}$$

where $\mathcal{D}(\tilde{\mu}_h) := \{\nu_h \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d) : \|\nu_h - \tilde{\mu}_h\|_M \le 1\}$.

In the following theorem, we show that we can find an approximate solution point such that (4.6) is satisfied with $\epsilon = O(\frac{1}{N})$ after N iterations of Algorithm 2. Therefore, a worst-case $O(\frac{1}{N})$ convergence rate is established for Algorithm 2.

Theorem 4.4 (Convergence rate in the ergodic sense)**.** *Let the sequence $\{\mu_h^{n+1}\}$ be generated by Algorithm 2 with $\rho \in (0, 2)$ under the condition (2.1). For any integer $N > 0$, let*

$$\tilde{\mu}_N = \frac{1}{N+1} \sum_{n=0}^{N} \bar{\mu}_h^n.$$

Then, we have

$$\big(F(\nu_h), \tilde{\mu}_N - \nu_h\big) \le \frac{1}{2\rho(N+1)} \|\nu_h - \mu_h^0\|_M^2, \ \forall\, \nu_h \in \mathcal{D}(\tilde{\mu}_N). \tag{4.7}$$

Proof. As $\rho \in (0, 2)$, it follows from (4.4) that

$$\begin{aligned} \big(F(\nu_h), \bar{\mu}_h^n - \nu_h\big) \le \frac{1}{2\rho}\big(\|\nu_h - \mu_h^n\|_M^2 - \|\nu_h - \mu_h^{n+1}\|_M^2\big), \\ \forall\, \nu_h \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d). \end{aligned} \tag{4.8}$$

Then, summarizing the above inequalities for $n = 0, \cdots, N$, we have

$$\big(F(\nu_h), \sum_{n=0}^{N} \bar{\mu}_h^n - (N+1)\nu_h\big) \le \frac{1}{2\rho}\big(\|\nu_h - \mu_h^0\|_M^2 - \|\nu_h - \mu_h^{N+1}\|_M^2\big), \tag{4.9}$$

which implies the result (4.7) immediately. □

This theorem shows a worst-case $O(\frac{1}{N})$ convergence rate in the ergodic sense for Algorithm 2. As [35], the "ergodic" sense means the approximate solution with an accuracy of $O(\frac{1}{N})$ is the average of all the N iterations generated by Algorithm 2.

4.2.2 Convergence Rate in a Nonergodic Sense

Indeed, we can derive a stronger worst-case $O(\frac{1}{N})$ convergence rate in a nonergodic sense. For this purpose, we first prove a lemma.

Lemma 4.1. *Let the sequence $\{\mu_h^{n+1}\}$ be generated by Algorithm 2 with $\rho \in (0, 2)$ under the condition (2.1). Then the sequence $\{\|\mu_h^n - \mu_h^{n+1}\|_M^2\}$ is monotonically non-increasing. That is, for any integer n, we have*

$$\|\mu_h^{n+1} - \mu_h^{n+2}\|_M^2 \le \|\mu_h^n - \mu_h^{n+1}\|_M^2. \tag{4.10}$$

Proof. It follows from (4.1) that $\bar{\mu}_h^n$ and $\bar{\mu}_h^{n+1}$, respectively, satisfy the following two inequalities:

$$\big(F(\bar{\mu}_h^n) + M(\bar{\mu}_h^n - \mu_h^n), \nu_h - \bar{\mu}_h^n\big) \geq 0, \ \forall\, \nu_h \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d), \tag{4.11}$$

and

$$\big(F(\bar{\mu}_h^{n+1}) + M(\bar{\mu}_h^{n+1} - \mu_h^{n+1}), \nu_h - \bar{\mu}_h^{n+1}\big) \geq 0, \ \forall\, \nu_h \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d). \tag{4.12}$$

Adding the above two inequalities together, choosing ν_h as $\bar{\mu}_h^{n+1}$ and $\bar{\mu}_h^n$ in (4.11) and (4.12), respectively, we obtain

$$\big(M\big((\mu_h^n - \bar{\mu}_h^n) - (\mu_h^{n+1} - \bar{\mu}_h^{n+1})\big), \bar{\mu}_h^n - \bar{\mu}_h^{n+1}\big) \geq 0. \tag{4.13}$$

Adding the term

$$\big(M\big((\mu_h^n - \bar{\mu}_h^n) - (\mu_h^{n+1} - \bar{\mu}_h^{n+1})\big), \big((\mu_h^n - \bar{\mu}_h^n) - (\mu_h^{n+1} - \bar{\mu}_h^{n+1})\big)\big)$$

to both sides of (4.13), and using (2.11), we get

$$\begin{aligned} &\big(M\big((\mu_h^n - \mu_h^{n+1}) - (\mu_h^{n+1} - \mu_h^{n+2})\big), \mu_h^n - \mu_h^{n+1}\big) \\ &\geq \frac{1}{\rho}\|(\mu_h^n - \mu_h^{n+1}) - (\mu_h^{n+1} - \mu_h^{n+2})\|_M^2. \end{aligned} \tag{4.14}$$

Then, applying the identity

$$\big(M(b-a), b\big) = \frac{1}{2}\big(\|b\|_M^2 - \|a\|_M^2 + \|a-b\|_M^2\big)$$

to (4.14) with $a = \mu_h^{n+1} - \mu_h^{n+2}$ and $b = \mu_h^n - \mu_h^{n+1}$, we obtain

$$\begin{aligned} \|\mu_h^{n+1} - \mu_h^{n+2}\|_M^2 &\leq \|\mu_h^n - \mu_h^{n+1}\|_M^2 - \frac{2-\rho}{\rho}\|(\mu_h^n - \mu_h^{n+1}) - (\mu_h^{n+1} - \mu_h^{n+2})\|_M^2 \\ &\leq \|\mu_h^n - \mu_h^{n+1}\|_M^2, \end{aligned} \tag{4.15}$$

which indicates that the sequence $\{\|\mu_h^n - \mu_h^{n+1}\|_M^2\}$ is monotonically non-increasing. □

Now, it is easy to show that the quantity $\|\mu_h^N - \mu_h^{N+1}\|_M^2$ is bounded by the order $O(\frac{1}{N})$, from which a stronger worst-case $O(\frac{1}{N})$ convergence rate in a nonergodic sense for Algorithm 2 is established. We summarize this result in the following theorem; its proof is based on the assertions in Lemma 4.1 and Theorem 4.1.

Theorem 4.5 (Convergence rate in a nonergodic sense)**.** *Let μ_h be a solution point of (1.6) and the sequence $\{\mu_h^{n+1}\}$ be generated by Algorithm 2 with $\rho \in (0, 2)$ under the condition (2.1). Then, for any integer $N > 0$, we have*

$$\|\mu_h^N - \mu_h^{N+1}\|_M^2 \leq \frac{\rho}{2-\rho}\frac{1}{(N+1)}\|\mu_h - \mu_h^0\|_M^2 = O(\frac{1}{N}). \tag{4.16}$$

Proof. It follows from (4.2) and (2.11) that

$$\|\mu_h^n - \mu_h^{n+1}\|_M^2 \leq \frac{\rho}{2-\rho}\Big(\|\mu_h^n - \mu_h\|_M^2 - \|\mu_h^{n+1} - \mu_h\|_M^2\Big). \tag{4.17}$$

Taking the sum of (4.17) from $n = 0$ to N, we obtain that

$$\sum_{n=0}^{N} \|\mu_h^n - \mu_h^{n+1}\|_M^2 \leq \frac{\rho}{2-\rho}\Big(\|\mu_h^0 - \mu_h\|_M^2 - \|\mu_h^{N+1} - \mu_h\|_M^2\Big). \tag{4.18}$$

Therefore, the result (4.16) follows immediately from (4.18) and the non-increasing monotonicity of the sequence $\{\|\mu_h^n - \mu_h^{n+1}\|_M^2\}$ in (4.10). □

Remark 4.1. *Recall that M defined in (2.10) is symmetric and positive definite under the condition (2.1). Therefore, if $\|\mu_h^N - \mu_h^{N+1}\|_M^2 = 0$, then we have $\mu_h^N = \mu_h^{N+1}$ and it follows from (2.11) that $\mu_h^N = \bar{\mu}_h^N$. Further, because of (4.1), we have*

$$\big(F(\mu_h^{N+1}), \nu_h - \mu_h^{N+1}\big) \geq 0, \quad \forall\, \nu_h \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d),$$

which means μ_h^{N+1} is a solution point of (1.6). In this sense, for the sequence $\{\mu_h^n\}$ generated by Algorithm 2, we can measure the accuracy of the iterate μ_h^{N+1} by the quantity $\|\mu_h^N - \mu_h^{N+1}\|_M^2$ and thus the conclusion in Theorem 4.5 means a worst-case $O(\frac{1}{N})$ convergence rate in a nonergodic sense for Algorithm 2.

5 Numerical Examples

In this section, we test the proposed Algorithms 1 and 2; and report some numerical results. In particular, we verify that Algorithm 2 is more efficient than Algorithm 1 because of the relaxation step (2.11). This conclusion is consistent to some observations in the PPA literature, e.g. [28]. All codes were written in C++ based on the finite element library `AFEPack` [40] and all experiments were run on a Linux desktop with i5-4570s Intel 2.9GHz four Processors and 8GB Memory. The triangular meshes used for the finite element discretization are generated by the package `EasyMesh` [48].

5.1 Examples

We focus on the case of (1.2) where the linear operator A arises from the Fredholm integral equations of the first kind in one and two dimensions. That is, A is given by

$$Au(x) := \int_\Omega k(x,\xi)u(\xi)\mathrm{d}\xi, \tag{5.1}$$

where the kernel $k(\cdot,\cdot) : \Omega \times \Omega \to \mathbb{R}$ is a given integrable function and Ω is a bounded domain in $\mathbb{R}^d$ with $d = 1$ or $d = 2$. It is well known (e.g., [29, 39]) that many problems of the type (5.1) are ill-posed and thus it is not practical to solve the equation (1.1) directly. We refer to, e.g., [9, 29, 33, 39, 55, 58, 60], for a variety of applications of (5.1) in remote sensing, indirect measurement, identification of distributed parameters, and so on. We will test the following examples.

Example 5.1. *First, let us define*

$$u(x) = \begin{cases} 1, & x \in [0.2, 0.4] \text{ or } [0.6, 0.8], \\ 0, & \text{otherwise}. \end{cases} \tag{5.2}$$

Then, choosing $k(x,\xi) = \mathrm{e}^{-\frac{|x-\xi|^2}{2\eta^2}}$ with $\eta = 0.05$ and $\Omega = (0,1)$ in (5.1), we obtain a specific example of (5.1) in $\mathbb{R}^1$ with

$$Au(x) := \frac{1}{\sqrt{2\pi}\eta}\int_0^1 \mathrm{e}^{-\frac{|x-\xi|^2}{2\eta^2}} u(\xi)\mathrm{d}\xi. \tag{5.3}$$

Now, we have the ill-posed equation (1.1) *with* $g(x) := Au(x)$ *where* A *is defined in* (5.3) *and the exact solution* $u(x)$ *is given in* (5.2). *We choose* $\alpha = 5.0 \times 10^{-4}$ *in* (1.2).

Example 5.2. *Our second example of* (5.1) *is the Fredholm integral equation of the first kind in two dimensions as follows*

$$Au(x) := \frac{1}{2\pi\eta^2} \int_\Omega \mathrm{e}^{-\frac{|x-\xi|^2}{2\eta^2}} u(\xi)\mathrm{d}\xi = g(x), \tag{5.4}$$

with $\Omega = (0,1)^2$ *and* $\eta = 0.05$. *Its exact solution* $u = \iota_{B(x_0,r)}$ *is the characteristic function of*

$$B(x_0, r) = \{x \in \Omega : |x - x_0| \le r,\ x_0 = (0.5, 0.5)\}$$

with $r = 0.2$. *That is, we solve the problem* (1.1) *with* $g(x) := Au(x)$ *defined in* (5.4). *We choose* $\alpha = 1.0 \times 10^{-3}$ *in* (1.2).

Example 5.3. *Last, we consider the example of* (5.1) *with* $\Omega = \{x : |x - x_0| \le 0.5,\ x_0 = (0.5, 0.5)\}$ *and the operator* A *is the case of* (5.4) *with* $\eta = 0.05$, *where the exact solution* $u(x)$ *is given by*

$$u(x) = \begin{cases} 1, & \big((\Omega_0 \cup \Omega_1) \setminus (\Omega_2 \cup \Omega_3)\big) \cup \Omega_4, \\ 0, & \textit{otherwise}, \end{cases}$$

with $\Omega_0 = \{(x_1, x_2) \in \Omega : x_1 \le 0.5\}$, $\Omega_1 = \{x \in \Omega : 0.1 \le |x - x_0| \le 0.25,\ x_0 = (0.5, 0.75)\}$, $\Omega_2 = \{x \in \Omega : |x - x_0| \le 0.25,\ x_0 = (0.5, 0.25)\}$, $\Omega_3 = \{x \in \Omega : |x - x_0| \le 0.1,\ x_0 = (0.5, 0.75)\}$ *and* $\Omega_4 = \{x \in \Omega : |x - x_0| \le 0.1,\ x_0 = (0.5, 0.25)\}$. *We take* $\alpha = 1.0 \times 10^{-4}$ *in the model* (1.2).

5.2 Algorithms to be Compared

5.2.1 Primal-Dual Methods Without Linearization

Instead of alleviating the subproblem (1.9) as the linearized surrogate (1.12), one can directly solve (1.8) by some standard solvers for systems of linear equations for the case where A^*A is computable. To see the advantages of the proposed algorithms, we compare them with the primal-dual scheme (1.8) where the u_h-subproblem is solved by the standard conjugate gradient (CG) and the generalized Minimum RESidual (GMRES) in [36], respectively. They are denoted by "PDCG" and "PDGMRES", respectively. Analogous to Section 2.2, we can see that the iterate $\mu_h^n = (u_h^n; p_h^n)$ generated by (1.8) can be characterized by the variational inequality

$$\big(F(\mu_h^{n+1}) + G(\mu_h^{n+1} - \mu_h^n), \nu_h - \mu_h^{n+1}\big) \ge 0, \quad \forall\, \nu_h \in \mathcal{S}^1(\mathcal{T}_h) \times \mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d)$$

where $F(\cdot)$ and ν_h are defined in (2.5); and

$$G := \begin{pmatrix} \frac{1}{\tau} I & \alpha\mathrm{div} \\ -\alpha\nabla & \frac{\sigma}{\tau} I \end{pmatrix}.$$

Similar to the condition that ensures the positive definiteness of M defined in (2.10), we restrict

$$\tau < \tau_1 := \sqrt{\sigma}/(\alpha\|\nabla\|) \tag{5.5}$$

to ensure the positive definiteness of G and thus the convergence of the scheme (1.8).

5.2.2 Primal-Dual-Dual Method

As mentioned in [15, Section 6.3.1], an alternative approach to overcome the necessity of solving the implicit equation (1.9) (i.e., the u_h-subproblem in (1.8)) is to additionally dualize the data fidelity term in (1.6). Omitting some details, we can derive that the resulting saddle-point reformulation of (1.2) is

(5.6)
$$\inf_{u_h\in\mathcal{S}^1(\mathcal{T}_h)}\sup_{\substack{p_h\in\mathcal{L}^0(\mathcal{T}_h)^d\\ q_h\in\mathcal{S}^1(\mathcal{T}_h)}}\Big\{\big(Au_h-g,q_h\big)-\frac{1}{2}\|q_h\|^2_{L^2(\Omega)}+\alpha\int_\Omega\nabla u_h\cdot p_h\,\mathrm{d}x-I_B(p_h)\Big\},$$

where u_h is the primal variable and both p_h and q_h are dual variables (hence, the name "primal-dual-dual" is used). Applying the primal-dual scheme in [15] to (5.6), we obtain the iterative scheme

(5.7)
$$\begin{cases} u_h^{n+1} = \underset{u_h\in\mathcal{S}^1(\mathcal{T}_h)}{\arg\min}\Big\{\big(Au_h-g,q_h^n\big)+\alpha\displaystyle\int_\Omega\nabla u_h\cdot p_h^n\,\mathrm{d}x+\frac{1}{2\tau}\|u_h-u_h^n\|^2_{L^2(\Omega)}\Big\},\\ \tilde{u}_h^{n+1} = 2u_h^{n+1}-u_h^n,\\ p_h^{n+1} = \underset{p_h\in\mathcal{L}^0(\mathcal{T}_h)^d}{\arg\max}\Big\{\alpha\displaystyle\int_\Omega\nabla\tilde{u}_h^{n+1}\cdot p_h\,\mathrm{d}x-I_B(p_h)-\frac{\sigma}{2\tau}\|p_h-p_h^n\|^2_{L^2(\Omega)}\Big\},\\ q_h^{n+1} = \underset{q_h\in\mathcal{S}^1(\mathcal{T}_h)}{\arg\max}\Big\{\big(A\tilde{u}_h^{n+1}-g,q_h\big)-\frac{1}{2}\|q_h\|^2_{L^2(\Omega)}-\frac{\sigma}{2\tau}\|q_h-q_h^n\|^2_{L^2(\Omega)}\Big\}. \end{cases}$$

Similar to the discussion in Section 2.2, the scheme (5.7) can be rewritten as the variational inequality

(5.8)
$$\big(F(\mu_h^{n+1})+H(\mu_h^{n+1}-\mu_h^n),\mu_h-\mu_h^{n+1}\big)\geq 0,\quad\forall\,\mu_h\in\mathcal{S}^1(\mathcal{T}_h)\times\mathcal{B}_1(\mathcal{L}^0(\mathcal{T}_h)^d)\times\mathcal{S}^1(\mathcal{T}_h),$$

where

$$\mu_h=\begin{pmatrix}u_h\\p_h\\q_h\end{pmatrix},\quad F(\mu_h)=\begin{pmatrix}-\alpha\mathrm{div}p_h+A^*q_h\\-\alpha\nabla u_h\\q_h-Au_h+g\end{pmatrix},\quad H=\begin{pmatrix}\frac{1}{\tau}I&\alpha\mathrm{div}&-A^*\\-\alpha\nabla&\frac{\sigma}{\tau}I&0\\-A&0&\frac{\sigma}{\tau}I\end{pmatrix}.$$

Thus, if the matrix form operator H defined above is guaranteed to be symmetric and positive definite, the convergence of the scheme (5.7) is ensured by analogy to the analysis in Section 4. This requires the condition

(5.9)
$$\tau<\tau_2:=\sqrt{\frac{\sigma}{2(\alpha^2\|\nabla\|^2+\|A\|^2)}}.$$

We shall compare the proposed algorithms with the PDDM (5.7) subject to the condition (5.9).

5.3 Implementation Details

When implementing algorithms, we use the stopping criterion

(5.10)
$$\frac{\|u_h^{n+1}-u_h^n\|_{L^2(\Omega)}}{\|u_h^{n+1}\|_{L^2(\Omega)}}\leq 1.0\times10^{-4},$$

and calculate

$$\text{L2err} := \|u_h^N - u_h\|_{L^2(\Omega)}, \tag{5.11}$$

to measure the accuracy of the iterate u_h^N generated by algorithms.

Note that the condition (2.1) to guarantee the convergence of Algorithms 1 and 2 can be rewritten as

$$\tau < \tau_3 := \frac{\sqrt{\sigma^2\|A\|^4 + 4\sigma\alpha^2\|\nabla\|^2} - \sigma\|A\|^2}{2\alpha^2\|\nabla\|^2}. \tag{5.12}$$

Below we list some parameters of the proposed algorithms and the mentioned algorithms to be compared.

- Algorithm 1 (Alg1) and Algorithm 2 (Alg2): $\tau = 0.95\tau_3$ where τ_3 is given by (5.12);
- PDCG and PDGMRES: $\tau = 0.95\tau_1$, where τ_1 is given by (5.5), with an accuracy of 1.0×10^{-6} for the CG and GMRES procedure in the u_h-subproblem of (1.8);
- PDDM (5.7): $\tau = 0.95\tau_2$, where τ_2 is given by (5.9).

Note that we choose the accuracy of 1.0×10^{-6} for the internal iterations of PDCG and PDGMRES because we have found in our experiments that lower accuracy may lead to divergence or very slow convergence and higher accuracy can hardly improve the performance. We take the initial iterate as $u_h^0 = g_h$, where g_h is the finite element discretization of $g(x)$ with noise and p_h^0 as the zero function.

5.4 Numerical Results

5.4.1 Results for Example 5.1

Some noise generated by $\delta\|g\|_{L^2(\Omega)}\text{randn}(x)$ is added to $g(x)$, where the values of $\text{randn}(x)$ over nodes of the mesh are sampled from the standard normal distribution and $\delta > 0$ is a noise-level parameter. To generate the finite element discretization, the mesh size over $[0, 1]$ is taken as 0.01. Accordingly, the condition number of the discretized matrix of A^*A in (1.9) is 1.1883×10^9. With our choices of the kernel function in (5.3) and the finite element discretization mesh, we have $\|A\|^2 = 0.0098$ and $1/\|\nabla\| = 3.0 \times 10^{-3}$. We fix $\sigma = 0.03$. Then, the values of τ for Algorithms 1 and 2, PDCG/PDGMRES and PDDM are 0.9822544, 0.987269, 0.6002106, respectively.

In Table 1, we report the iteration numbers ("N"), computing time in seconds ("CPU(s)") and L^2 errors ("L2err") defined in (5.11) for Example 5.1. Recall that Algorithm 1 is the special case of Algorithm 2 with $\rho = 1.0$. Several other choices of ρ are also tested. We test the cases with noise levels $\delta = 20\%, 10\%, 5\%$ and 1%. According to the data in this table, we see that the proposed linearized primal-dual schemes work well for Example 5.1 and they are both faster than PDDM. In particular, we see that Algorithm 2 with $\rho \in (1, 2)$ is usually faster than the special case of $\rho = 1$, i.e., Algorithm 1. Also, we see that PDCG and PDGMRES perform nearly the same as Algorithm 1 for the case where the size of discretized matrix of A^*A is 101×101. It will be shown later that they are less efficient than Algorithm 1 for the case where the size of the discretized matrix of A^*A is larger.

In Figure 1, for Example 5.1 with the tested noise levels, we plot the graphs of u_h, Au_h, g_h and the numerical solution u_h^N generated by Algorithm 2 with $\rho = 1.6$. The curve of the exact solution given in (5.2) is also plotted for comparison. We see that the exact solution of

Example 5.1 can be very accurately approximated; thus the effectiveness of the model (1.2) with TV regularization is also well justified. Figure 2 plots the convergence rate measured by $\|\mu_h^N - \mu_h^{N+1}\|_M^2$ of Algorithms 1 and 2 with $\rho = 1.6$ for Example 5.1 with different noise levels. It is shown that practically the proposed algorithms can perform more favorably than the worst-case estimate of the theoretical convergence rate derived in Section 4. In Figure 3, we plot the convergence order of the error $E(u_h^N) - E(u)$ and $\|u_h^N - u\|_{L^2(\Omega)}$ for the sequence generated by Algorithm 1 for Example 5.1 with noise level 0.01, where u is an approximation of the true solution of (1.2) generated by a finer mesh with $h = 1.25 \times 10^{-3}$.

Table 1: Iteration numbers N, computing time in seconds and L^2 errors by Algorithm 1 (Alg1), Algorithm 2 (Alg2) with $\rho = 0.4, 0.6, \cdots, 1.8$, and PDCG, PDGMRES, PDDM for Example 5.1 with noise levels $\delta = 20\%, 10\%, 5\%$ and 1%.

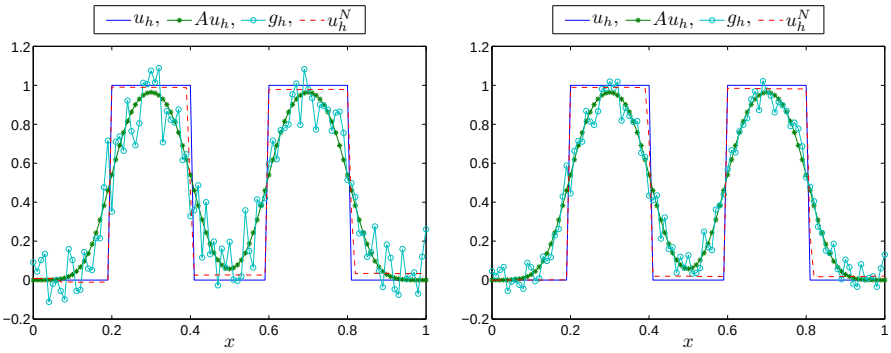
		$\delta = 20\%$			$\delta = 10\%$			$\delta = 5\%$			$\delta = 1\%$		
		N	CPU(s)	L2err	N	CPU(s)	L2err	N	CPU(s)	L2err	N	CPU(s)	L2err
Alg1		551	0.81	0.0572	502	0.72	0.0324	524	0.76	0.0197	528	0.75	0.0131
Alg2	ρ=0.4	709	1.01	0.0615	823	1.18	0.0358	927	1.33	0.0251	1064	1.54	0.0200
	ρ=0.6	765	1.09	0.0581	796	1.16	0.0319	685	0.98	0.0216	806	1.17	0.0150
	ρ=0.8	655	0.94	0.0575	603	0.87	0.0319	629	0.90	0.0194	652	0.93	0.0133
	ρ=1.2	462	0.67	0.0572	424	0.61	0.0325	456	0.65	0.0202	455	0.65	0.0126
	ρ=1.4	399	0.57	0.0571	365	0.53	0.0324	396	0.57	0.0204	403	0.58	0.0123
	ρ=1.6	**363**	0.52	0.0567	**339**	0.51	0.0328	**369**	0.55	0.0208	**364**	0.52	0.0124
	ρ=1.8	429	0.61	0.0572	417	0.61	0.0318	412	0.61	0.0204	413	0.60	0.0139
PDCG		554	0.84	0.0572	502	0.76	0.0323	524	0.79	0.0197	527	0.82	0.0131
PDGMRES		554	0.74	0.0572	502	0.70	0.0323	524	0.70	0.0197	527	0.72	0.0131
PDDM		839	1.39	0.0576	799	1.28	0.0319	828	1.33	0.0194	835	1.32	0.0138

5.4.2 Results for Example 5.2

In our experiments, the domain Ω is partitioned by the regular triangular mesh with 1046 nodes and 1990 elements. The partition is shown in Figure 4(a). In Figures 4(b) and (c), we plot the graphs of u_h and Au_h. The noise $\delta\|g\|_{L^2(\Omega)}\mathrm{randn}(x)$ is added to $g(x)$, where $\mathrm{randn}(x)$ is the standard normal distribution over the finite element mesh and $\delta > 0$ is a noise-level parameter. Accordingly, the condition number of the discretized matrix of A^*A in (1.9) is 5.4031×10^{10}. Note that for the kernel function and finite element mesh for Example 5.2, we have $\|A\|^2 = 0.002$ and $1/\|\nabla\| = 3.5 \times 10^{-3}$. We fix $\sigma = 0.12$. Then, the values of τ for Algorithms 1 and 2, PDCG/PDGMRES and PDDM are 1.150418, 1.151814, 0.8046579, respectively.

The iteration numbers ("N"), computing time in seconds ("CPU(s)") and L^2 errors ("L2err") are reported in Table 2 for Example 5.2 with different noise levels $\delta = 20\%, 10\%, 5\%$ and 1%. The efficiency of the proposed algorithms, and in particular, the acceleration effectiveness of Algorithm 2, are verified again by the data in Table 2. It should be noted that PDCG and PDGMRES require much more computing time compared with Algorithm 1 even though their iteration numbers (for the outer iteration) are almost the same. Again, we see the superiority of Algorithm 2 over Algorithm 1 with appropriate choices of ρ; and their superiority to PDDM.

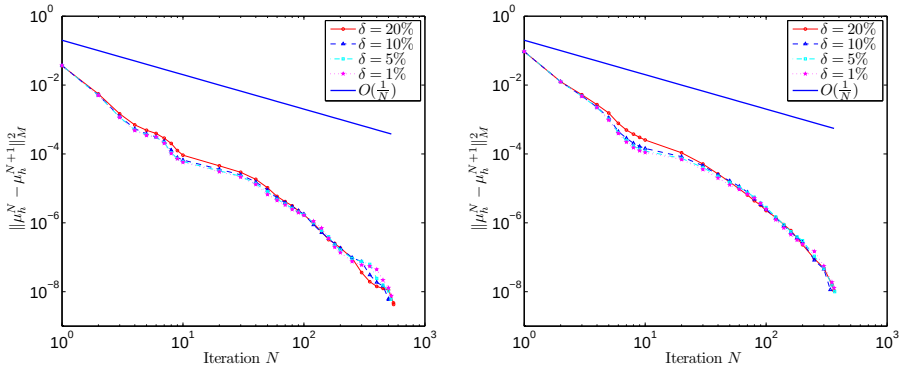
In Figure 5, we plot the graphs of g_h and the corresponding numerical solution u_h^N of Example 5.2 with different noise levels for Algorithm 2 with $\rho = 1.2$. Again, the plots in this figure show that the exact solution of Example 5.2 can be very accurately approximated. Thus, the model (1.2) with TV regularization is well verified. In Figure 6, we demonstrate the convergence rate of Algorithms 1 and 2 for Example 5.2; and in Figure 7 we plot the



(a) noise level $\delta = 20\%$ (b) noise level $\delta = 10\%$

(c) noise level $\delta = 5\%$ (d) noise level $\delta = 1\%$

Figure 1: Graphs of u_h, Au_h, g_h and the numerical solution u_h^N of Example 5.1 for different noise levels by Algorithm 2 with $\rho = 1.6$. (u_h: exact solution; g_h: data with noise; u_h^N: output numerical solution)



(a) Algorithm 1 (b) Algorithm 2 with $\rho = 1.6$

Figure 2: Convergence rate measured by $\|\mu_h^N - \mu_h^{N+1}\|_M^2$ of Algorithm 1 and Algorithm 2 with $\rho = 1.6$ for Example 5.1 with different noise levels.

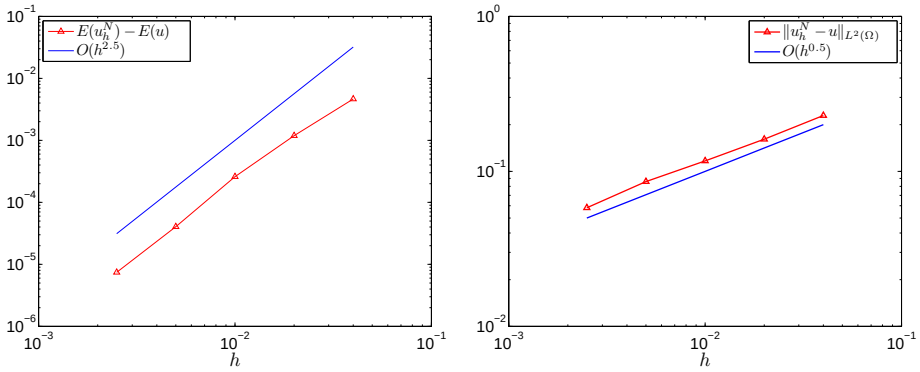
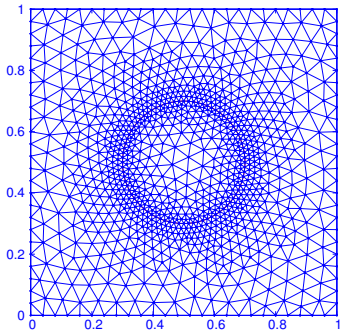
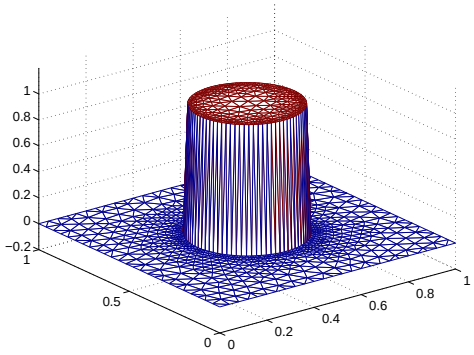


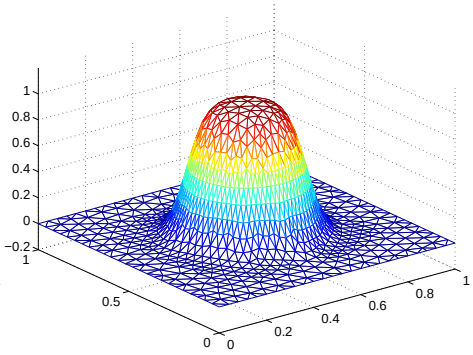
Figure 3: Convergence order of errors $E(u_h^N) - E(u)$ and $\|u_h^N - u\|_{L^2(\Omega)}$ by Algorithm 1 for Example 5.1 with noise level 0.01.



(a) mesh



(b) u_h



(c) Au_h

Figure 4: Triangular mesh over Ω and the graphs of u_h and Au_h for Example 5.2.

convergence order of errors $E(u_h^N) - E(u)$ and $\|u_h^N - u\|_{L^2(\Omega)}$ by Algorithm 1 for Example 5.2 with noise level 0.01.

Table 2: Iteration numbers N, computing time in seconds and L^2 errors by Algorithm 1 (Alg1), Algorithm 2 (Alg2) with $\rho = 0.4, 0.6, \cdots, 1.8$ and PDCG, PDGMRES, PDDM for Example 5.2 with noise levels $\delta = 20\%, 10\%, 5\%$ and 1%.

		$\delta = 20\%$			$\delta = 10\%$			$\delta = 5\%$			$\delta = 1\%$		
		N	CPU(s)	L2err	N	CPU(s)	L2err	N	CPU(s)	L2err	N	CPU(s)	L2err
Alg1		310	8.68	0.0243	326	9.29	0.0137	252	7.22	0.0107	255	7.21	0.0099
Alg2	ρ=0.4	454	12.62	0.0248	420	11.68	0.0140	412	11.41	0.0119	399	11.18	0.0110
	ρ=0.6	357	10.09	0.0242	319	8.91	0.0135	323	9.04	0.0113	316	8.86	0.0107
	ρ=0.8	342	9.54	0.0242	371	10.47	0.0135	276	7.81	0.0109	261	7.41	0.0106
	ρ=1.2	**297**	8.35	0.0244	290	8.27	0.0137	**230**	6.54	0.0105	**235**	6.67	0.0096
	ρ=1.4	314	8.79	0.0242	**274**	7.73	0.0138	262	7.40	0.0096	268	7.71	0.0088
	ρ=1.6	336	9.38	0.0240	298	8.41	0.0137	302	8.48	0.0093	279	7.85	0.0084
	ρ=1.8	450	12.49	0.0240	421	11.81	0.0139	416	11.52	0.0095	411	11.38	0.0079
PDCG		314	20.85	0.0243	327	21.26	0.0136	255	16.06	0.0106	257	16.07	0.0098
PDGMRES		315	22.10	0.0242	324	22.11	0.0136	253	16.74	0.0106	256	16.73	0.0098
PDDM		451	14.56	0.0243	444	14.09	0.0136	388	12.64	0.0105	386	12.36	0.0096

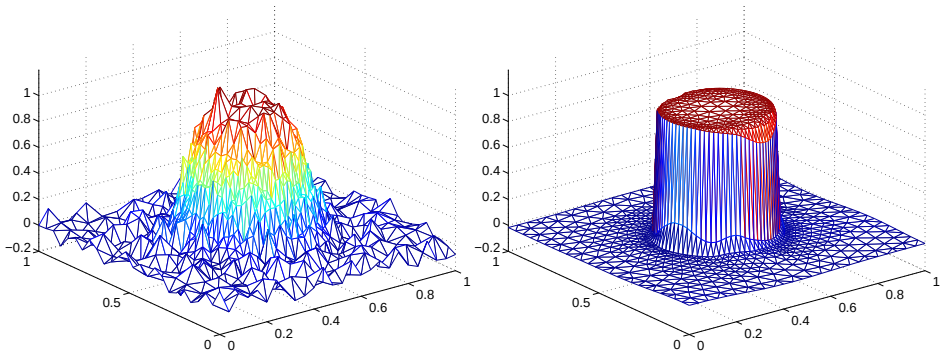
5.4.3 Results for Example 5.3

As shown in Figure 9(a), the appearance of $u(x)$ over the triangular mesh with 2647 nodes and 5092 elements is the Tai-Chi diagram, and its blurred image is shown in Figure 9(b). So, solving (1.2) can be regarded as a deblurring problem for the Tai-Chi image. Accordingly, the condition number of the discretized matrix of A^*A in (1.9) is 1.1013×10^{11}. For the kernel function and finite element mesh for Example 5.3 , we have $\|A\|^2 = 8.4527 \times 10^{-4}$ and $1/\|\nabla\| = 5.0 \times 10^{-4}$. We fix $\sigma = 4.0 \times 10^{-3}$. Then, the values of τ for Algorithms 1 and 2, PDCG/PDGMRES and PDDM are 0.3003762, 0.3004164, 0.2102169, respectively.

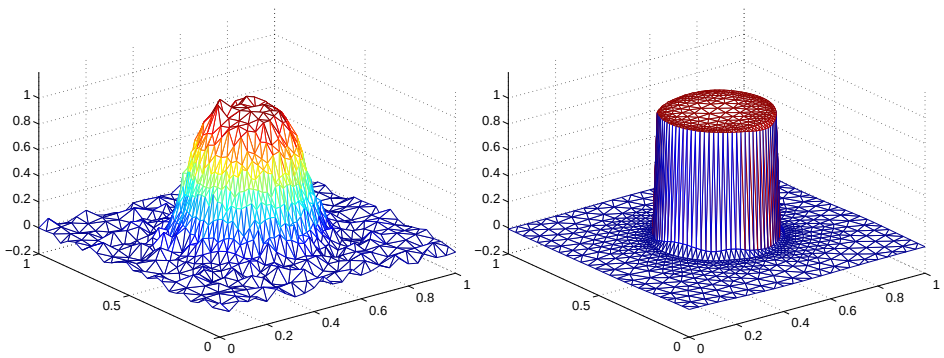
In Table 3, we report some numerical results for Example 5.3. The data clearly shows the superiority of Algorithms 1 and 2. In Figure 8, we plot the blurred images of u_h with different noise levels and the corresponding output iteration u_h^N obtained by Algorithm 2 with $\rho = 1.4$. We see that the exact solution can be well approximated by the model (1.2) and iteratively approached by the proposed algorithms. In particular, the edges of the original image can be well recovered. The convergence rates of Algorithm 1 and Algorithm 2 with $\rho = 1.4$ for Example 5.3 are plotted in Figure 10.

Table 3: Iteration numbers N, computing time in seconds and L^2 errors by Algorithm 1 (Alg1), Algorithm 2 (Alg2) with $\rho = 0.4, 0.6, \cdots, 1.8$ and PDCG, PDGMRES, PDDM for Example 5.3 with noise levels $\delta = 20\%, 10\%, 5\%$ and 1%.

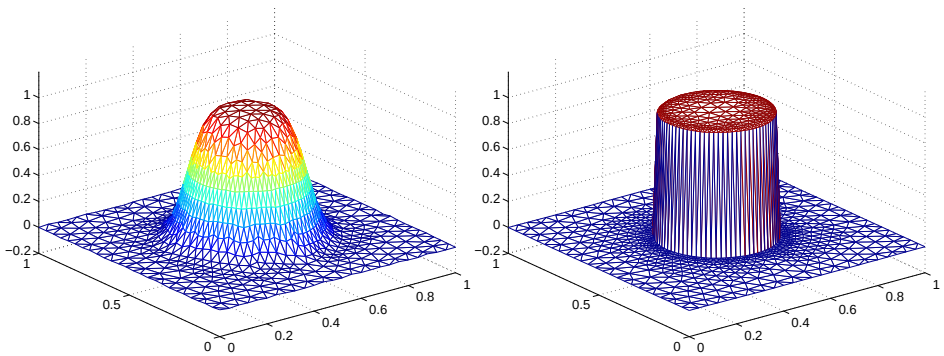
		$\delta = 20\%$			$\delta = 10\%$			$\delta = 5\%$			$\delta = 1\%$		
		N	CPU(s)	L2err	N	CPU(s)	L2err	N	CPU(s)	L2err	N	CPU(s)	L2err
Alg1		745	62.61	0.1086	1045	84.43	0.0749	1193	92.51	0.0639	1316	101.70	0.0578
Alg2	ρ=0.4	1013	81.49	0.1145	1199	96.71	0.0912	1314	101.45	0.0839	1331	103.23	0.0819
	ρ=0.6	894	72.24	0.1117	1179	96.79	0.0830	1270	99.20	0.0748	1348	105.52	0.0709
	ρ=0.8	810	65.71	0.1099	1111	90.16	0.0782	1237	96.52	0.0683	1376	106.39	0.0625
	ρ=1.2	709	58.36	0.1074	988	79.84	0.0726	**1120**	87.46	0.0613	**1280**	100.30	0.0540
	ρ=1.4	**689**	56.38	0.1064	**954**	78.23	0.0706	1139	89.92	0.0578	1287	98.91	0.0505
	ρ=1.6	705	58.59	0.1053	956	77.88	0.0686	1179	93.26	0.0548	1348	105.37	0.0475
	ρ=1.8	805	66.58	0.1038	1076	88.36	0.0657	1382	110.87	0.0512	1527	119.59	0.0449
PDCG		746	321.91	0.1086	1042	439.85	0.0751	1191	503.44	0.0641	1314	557.73	0.0580
PDGMRES		747	312.52	0.1087	1053	425.05	0.0750	1200	487.84	0.0639	1320	539.16	0.0576
PDDM		936	93.18	0.1092	1337	128.68	0.0774	1545	140.82	0.0668	1662	152.03	0.0617



(a) g_h with $\delta = 20\%$ (b) Numerical solution u_h^N with $\delta = 20\%$



(c) g_h with $\delta = 10\%$ (d) Numerical solution u_h^N with $\delta = 10\%$

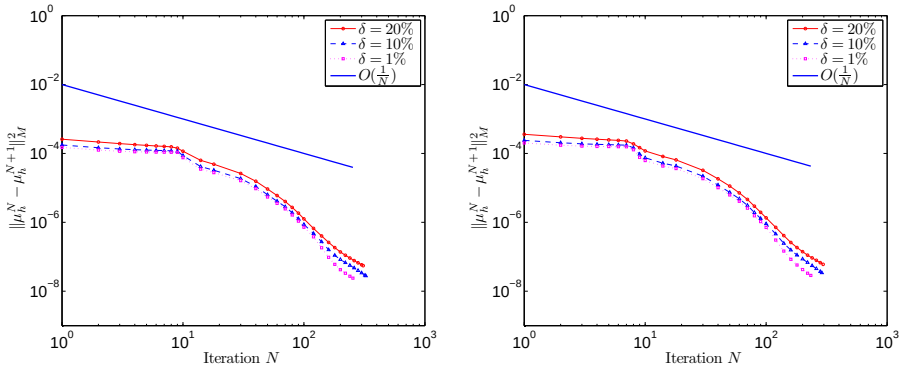


(e) g_h with $\delta = 1\%$ (f) Numerical solution u_h^N with $\delta = 1\%$

Figure 5: Graphs of g_h with noise levels $\delta = 20\%, 10\%, 1\%$ and the corresponding numerical solution u_h^N of Example 5.2 by Algorithm 2 with $\rho = 1.2$.

6 Conclusions

In this paper, we studied the primal-dual approach to the saddle-point reformulation of a linear inverse problem with the total variation regularization and finite element discretization. We modified the original primal-dual scheme by linearizing the data-fidelity term at each iteration and thus obtained easier subproblems. A linearized primal-dual method was thus



(a) Algorithm 1 (b) Algorithm 2 with $\rho = 1.2$

Figure 6: Convergence rate measured by $\|\mu_h^N - \mu_h^{N+1}\|_M^2$ of Algorithm 1 and Algorithm 2 with $\rho = 1.2$ for Example 5.2 with different noise levels.

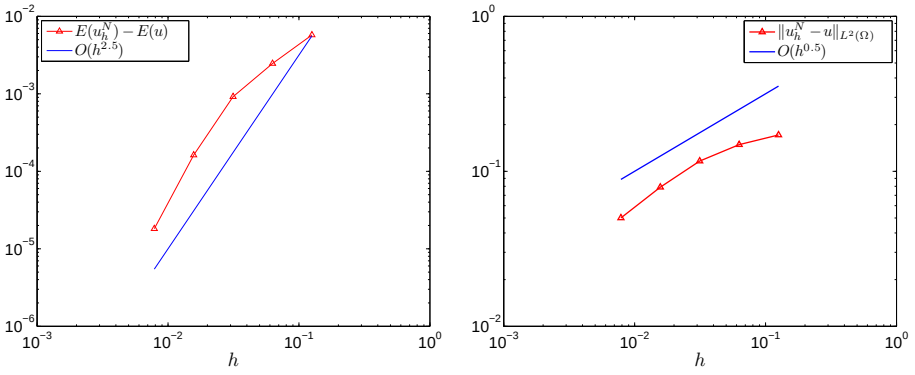
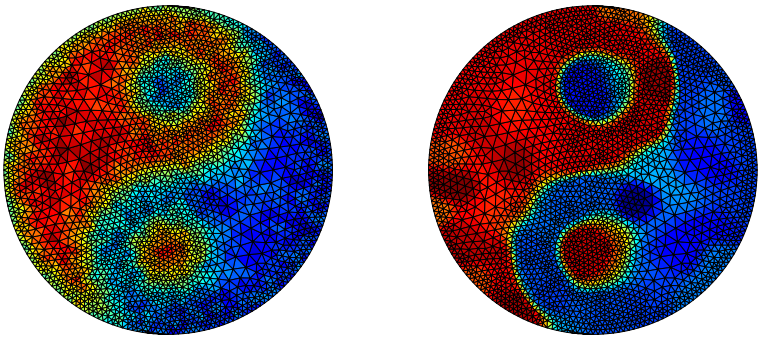
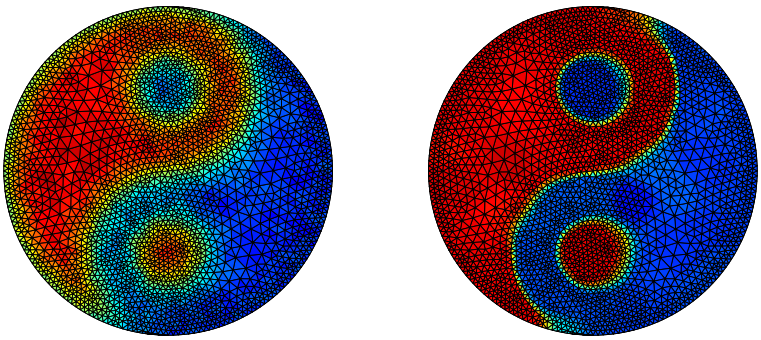


Figure 7: Convergence order of errors $E(u_h^N) - E(u)$ and $\|u_h^N - u\|_{L^2(\Omega)}$ by Algorithm 1 for Example 5.2 with noise level 0.01.

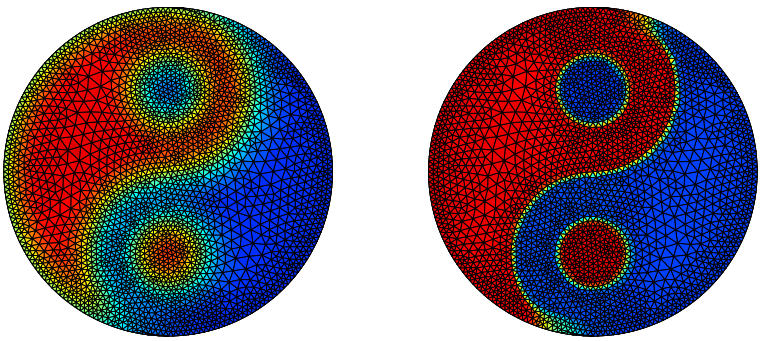
proposed. A relaxed version of this linearized primal-dual method was also studied, mainly inspired by the fact that it is indeed an application of the proximal point algorithm. We established the global convergence and estimated the worst-case convergence rate measured by the iteration complexity for both algorithms. Their efficiency was verified by preliminary numerical results. The linearized primal-dual method has the main advantage that the linearized surrogate subproblems are much easier and inner iteration is completely avoided for each linearized subproblem. That is, it can be regarded as a special approximate version of the original primal-dual method which linearizes the data-fidelity term at each iteration, and each iterate is the exact solution of the linearized surrogate. An alternative is keeping the hard subproblem completely at each iteration of the original primal-dual method and adopting a certain inner iteration process to solve the subproblem iteratively — no surrogate subproblem is generated and an inner iteration is necessary at each step. We partially studied this alternative empirically and it seems that algorithms based on this approach are less efficient than our linearized primal-dual approach. It is certainly interesting to conduct more serious research on how to design an easily implementable stopping criterion to terminate the inner iteration; how to balance the errors accumulated in the inner iteration and



(a) g_h with $\delta = 20\%$ (b) Numerical solution u_h^N with $\delta = 20\%$



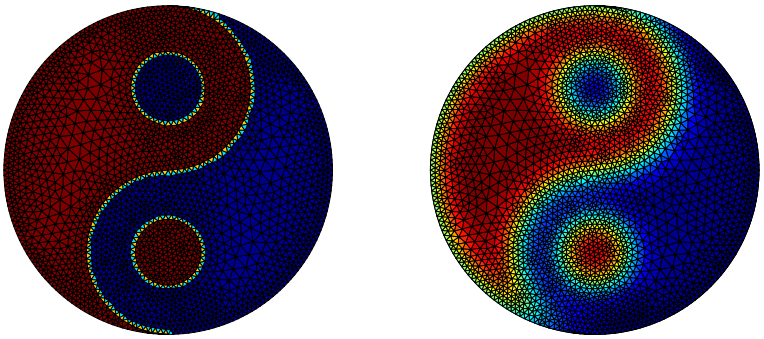
(c) g_h with $\delta = 10\%$ (d) Numerical solution u_h^N with $\delta = 10\%$



(e) g_h with $\delta = 1\%$ (f) Numerical solution u_h^N with $\delta = 1\%$

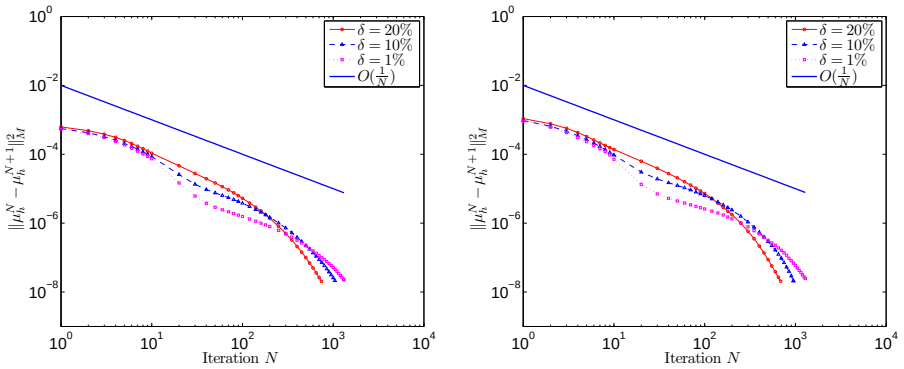
Figure 8: Graphs of g_h with noise levels $\delta = 20\%, 10\%, 1\%$ and the corresponding numerical solution u_h^N of Example 5.3 by Algorithm 2 with $\rho = 1.4$.

the accuracy of the outer iteration; and how to prove the convergence rigorously for these algorithms.



(a) u_h (b) Au_h

Figure 9: Image of Tai-Chi diagram and its blurred image for Example 5.3.



(a) Algorithm 1 (b) Algorithm 2 with $\rho = 1.4$

Figure 10: Convergence rate measured by $\|\mu_h^N - \mu_h^{N+1}\|_M^2$ of Algorithm 1 and Algorithm 2 with $\rho = 1.4$ for Example 5.3 with different noise levels.

References

[1] R. ACAR AND C. R. VOGEL, *Analysis of bounded variation penalty methods for ill-posed problems*, Inverse Problems, 10 (1994), pp. 1217–1229.

[2] W. K. ALLARD, *Total variation regularization for image denoising, I. Geometric theory*, SIAM J. Math. Anal., 39 (2008), pp. 1150–1190.

[3] L. AMBROSIO, N. FUSCO, AND D. PALLARA, *Functions of bounded variation and free discontinuity problems*, Clarendon Press, Oxford, 2000.

[4] K. J. ARROW, L. HURWICZ, AND H. UZAWA, *Studies in linear and non-linear programming*, Stanford University Press, Stanford, 1958.

[5] H. ATTOUCH, G. BUTTAZZO, AND G. MICHAILLE, *Variational analysis in Sobolev and BV spaces: Applications to PDEs and Optimization*, SIAM/MPS, Philadelphia, PA, 2006.

[6] G. AUBERT AND P. KORNPROBST, *Mathematical problems in image processing: Partial differential equations and the calculus of variations*, Springer, New York, second ed., 2006.

[7] S. BARTELS, *Total variation minimization with finite elements: convergence and iterative solution*, SIAM J. Numer. Anal., 50 (2012), pp. 1162–1180.

[8] ———, *Numerical Methods for Nonlinear Partial Differential Equations*, Springer International Publishing, AG Switzerland, 2015.

[9] J. BAUMEISTER, *Stable Solution of Inverse Problems*, Friedr. Vieweg & Sohn, Braunschweig, 1987.

[10] K. BREDIES AND H. SUN, *Preconditioned Douglas-Rachford splitting methods for convex-concave saddle-point problems*, SIAM J. Numer. Anal., 53 (2015), pp. 421–444.

[11] S. C. BRENNER AND L. R. SCOTT, *The mathematical theory of finite element methods*, Springer, New York, third ed., 2008.

[12] E. CASAS, K. KUNISCH, AND C. POLA, *Regularization by functions of bounded variation and applications to image enhancement*, Appl. Math. Optim., 40 (1999), pp. 229–257.

[13] V. CASELLES, A. CHAMBOLLE, AND M. NOVAGA, *The discontinuity set of solutions of the TV denoising problem and some extensions*, Multiscale Model. Simul., 6 (2007), pp. 879–894.

[14] A. CHAMBOLLE AND P.-L. LIONS, *Image recovery via total variation minimization and related problems*, Numer. Math., 76 (1997), pp. 167–188.

[15] 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.

[16] T. F. CHAN AND S. ESEDOḠLU, *Aspects of total variation regularized L^1 function approximation*, SIAM J. Appl. Math., 65 (2005), pp. 1817–1837.

[17] T. F. CHAN AND X.-C. TAI, *Identification of discontinuous coefficients in elliptic problems using total variation regularization*, SIAM J. Sci. Comput., 25 (2003), pp. 881–904.

[18] G. CHAVENT AND K. KUNISCH, *Regularization of linear least squares problems by total bounded variation*, ESAIM Control Optim. Calc. Var., 2 (1997), pp. 359–376.

[19] Z. CHEN AND J. ZOU, *An augmented lagrangian method for identifying discontinuous parameters in elliptic systems*, SIAM J. Control Optim., 37 (1999), pp. 892–910.

[20] C. CLASON, B. JIN, AND K. KUNISCH, *A duality-based splitting method for ℓ^1-TV image restoration with automatic regularization parameter choice*, SIAM J. Sci. Comput., 32 (2010), pp. 1484–1505.

[21] C. CLASON AND K. KUNISCH, *A duality-based approach to elliptic control problems in non-reflexive Banach spaces*, ESAIM Control Optim. Calc. Var., 17 (2011), pp. 243–266.

[22] R. A. DEVORE AND G. G. LORENTZ, *Constructive approximation*, Springer-Verlag, Berlin, 1993.

[23] D. C. DOBSON AND F. SANTOSA, *An image-enhancement technique for electrical impedance tomography*, Inverse Problems, 10 (1994), pp. 317–334.

[24] ———, *Recovery of blocky images from noisy and blurred data*, SIAM J. Appl. Math., 56 (1996), pp. 1181–1198.

[25] H. W. ENGL, M. HANKE, AND A. NEUBAUER, *Regularization of inverse problems*, Kluwer Academic Publishers Group, Dordrecht, 1996.

[26] E. ESSER, X. ZHANG, AND T. F. CHAN, *A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science*, SIAM J. Imaging Sci., 3 (2010), pp. 1015–1046.

[27] F. FACCHINEI AND J.-S. PANG, *Finite-dimensional variational inequalities and complementarity problems. Vol. I*, Springer-Verlag, New York, 2003.

[28] E. G. GOL'SHTEIN AND N. V. TRET'YAKOV, *Modified Lagrangians in convex programming and their generalizations*, in Point-to-Set Maps and Mathematical Programming, P. Huard, ed., Springer Berlin Heidelberg, 1979, pp. 86–97.

[29] C. W. GROETSCH, *The theory of Tikhonov regularization for Fredholm equations of the first kind*, Pitman (Advanced Publishing Program), Boston, MA, 1984.

[30] G. Y. Gu, B. S. He, and X. M. Yuan, *Customized proximal point algorithms for linearly constrained convex minimization and saddle-point problems: a unified approach*, Comput. Optim. Appl., 59 (2013), pp. 135–161.

[31] S. Gutman, *Identification of discontinuous parameters in flow equations*, SIAM J. Control Optim., 28 (1990), pp. 1049–1060.

[32] H. Gzyl and Y. Velásquez, *Linear Inverse Problems: The Maximum Entropy Connection*, World Scientific Publishing Co. Pte. Ltd., Singapore, 2011.

[33] P. C. Hansen, *Discrete inverse problems: Insight and algorithms*, SIAM, Philadelphia, PA, 2010.

[34] B. S. He and X. M. Yuan, *Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective*, SIAM J. Imaging Sci., 5 (2012), pp. 119–149.

[35] ———, *On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method*, SIAM J. Numer. Anal., 50 (2012), pp. 700–709.

[36] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995.

[37] Y. L. Keung and J. Zou, *Numerical identifications of parameters in parabolic systems*, Inverse Problems, 14 (1998), pp. 83–100.

[38] A. Kirsch, *An introduction to the mathematical theory of inverse problems*, Springer, New York, second ed., 2011.

[39] R. Kress, *Linear integral equations*, Springer, New York, third ed., 2014.

[40] R. Li, *AFEPack*. http://dsec.pku.edu.cn/~rli/AFEPack-snapshot.tar.gz.

[41] J. L. Lions, *Optimal Control of Systems Governed by Partial Differential Equations*, Springer, Berlin, 1971.

[42] B. Martinet, *Régularisation dinéquations variationnelles par approximations successives*, Rev. Franç. Inform. Rech. Opér., 4 (1970), pp. 154–158.

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

[44] D. Mumford and J. Shah, *Optimal approximations by piecewise smooth functions and associated variational problems*, Comm. Pure Appl. Math., 42 (1989), pp. 577–685.

[45] A. S. Nemirovsky and D. B. Yudin, *Problem complexity and method efficiency in optimization*, John Wiley & Sons, Inc., New York, 1983.

[46] Y. Nesterov, *Gradient methods for minimizing composite functions*, Math. Program., 140 (2013), pp. 125–161.

[47] Y. E. Nesterov, *A method for solving the convex programming problem with convergence rate $O(1/k^2)$*, Dokl. Akad. Nauk SSSR, 269 (1983), pp. 543–547. (In Russian. Translated in Soviet Math. Dokl., 27 (1983), pp. 372-376.).

[48] B. Niceno, *EasyMesh*. http://web.mit.edu/easymesh_v1.4/www/easymesh.html.

[49] N. Parikh and S. Boyd, *Proximal algorithms*, Foundations and Trends in Optimization, 1 (2014), pp. 127–239.

[50] T. Pock and A. Chambolle, *Diagonal preconditioning for first order primal-dual algorithms in convex optimization*, in Computer Vision (ICCV), 2011 IEEE International Conference on, Nov 2011, pp. 1762–1769.

[51] W. Ring, *Structural properties of solutions to total variation regularization problems*, ESAIM Math. Model. Numer. Anal., 34 (2000), pp. 799–810.

[52] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.

[53] D. STRONG AND T. CHAN, *Edge-preserving and scale-dependent properties of total variation regularization*, Inverse Problems, 19 (2003), pp. S165–S187.

[54] W. Y. TIAN AND X. M. YUAN, *Convergence analysis of primal-dual based methods for total variation minimization with finite element approximation*, submited, (2014).

[55] A. N. TIKHONOV AND V. Y. ARSENIN, *Solutions of Ill-Posed Problems*, Wiley, New York, 1977. (translated from the Russian).

[56] A. N. TIKHONOV, A. V. GONCHARSKY, V. V. STEPANOV, AND A. G. YAGOLA, *Numerical Methods for the Solution of Ill-Posed Problems*, Kluwer, Dordrecht, 1995. (Translated from the 1990 Russian original by R. A. M. Hoksbergen and revised by the authors).

[57] F. TRÖLTZSCH, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, American Mathematics Society, Providence, RI, 2010.

[58] C. R. VOGEL, *Computational methods for inverse problems*, SIAM, Philadelphia, PA, 2002.

[59] J. WANG AND B. J. LUCIER, *Error bounds for finite-difference methods for Rudin-Osher-Fatemi image smoothing*, SIAM J. Numer. Anal., 49 (2011), pp. 845–868.

[60] G. M. WING, *A primer on integral equations of the first kind: The problem of deconvolution and unfolding*, SIAM, Philadelphia, PA, 1991.

[61] M. ZHU AND T. F. CHAN, *An efficient primal-dual hybrid gradient algorithm for total variation image restoration*, CAM Report 08-34, UCLA, Los Angeles, CA, 2008.

[62] W. P. ZIEMER, *Weakly differentiable functions: Sobolev spaces and functions of bounded variation*, Springer-Verlag, New York, 1989.