SIAM J. NUMER. ANAL.
Vol. 61, No. 5, pp. 2062–2083

A UNIFORM PRECONDITIONER FOR A NEWTON ALGORITHM FOR TOTAL VARIATION MINIMIZATION AND MINIMUM-SURFACE PROBLEMS*

XUE-CHENG TAI†, RAGNAR WINTHER‡, XIAODI ZHANG§, AND WEIYING ZHENG¶

Abstract. Solution methods for the nonlinear PDE of the Rudin–Osher–Fatemi (ROF) and minimum-surface models are fundamental for many modern applications. Many efficient algorithms have been proposed. First-order methods are common. They are popular due to their simplicity and easy implementation. Some second-order Newton-type iterative methods have been proposed, such as Chan–Golub–Mulet method. In this paper, we propose a new Newton–Krylov solver for primal-dual finite element discretization of the ROF model and the minimum-surface model. The method is so simple that we just need to use some diagonal preconditioners during the iterations. Theoretically, the proposed preconditioners are further proved to be robust and optimal with respect to the mesh size, the penalization parameter, the regularization parameter, and the iterative step; essentially, it is a parameter-independent preconditioner. We first discretize the primal-dual system by using mixed finite element methods and then linearize the discrete system by Newton's method. Exploiting the well-posedness of the linearized problem on appropriate Sobolev spaces equipped with proper norms, we propose block diagonal preconditioners for the corresponding system solved with the minimum residual method. Numerical results are presented to support the theoretical results.

Key words. ROF model, primal-dual, finite element method, Newton method, block preconditioners

MSC codes. 65M60, 65M12

DOI. 10.1137/22M1512776

1. Introduction.

Image restoration is a fundamental and challenging task in image processing. A surge of research has been done in variational and PDE-based approaches. The Rudin–Osher–Fatemi (ROF) model, due to Rudin, Osher, and Fatemi [37], is one of the most successfully and widely used mathematical models. Given an image $f : \Omega \subset \mathbb{R}^d \mapsto \mathbb{R}$, $d = 1, 2, 3$, the ROF model is trying to solve the following minimization problem:

$$\min_{v \in \mathrm{BV}(\Omega)} \left\{ E(v) = \int_\Omega \alpha |\nabla v| \mathrm{d}\boldsymbol{x} + \frac{1}{2} \int_\Omega (v - f)^2 \mathrm{d}\boldsymbol{x} \right\}, \tag{1.1}$$

*Received by the editors August 1, 2022; accepted for publication (in revised form) March 24, 2023; published electronically September 14, 2023.

https://doi.org/10.1137/22M1512776

Funding: The work of the first author was supported by the National Natural Science Foundation of China (NSFC)/RGC projects N-CityU214/19, and NORCE kompetanseoppbygging program. The work of the third author was partially supported by NSFC grant 12201575 and China Postdoctoral Science Foundation grant 2022M722878. The forth author was supported in part by China NSF grants 11831016 and 12226354 and by the National Science Fund for Distinguished Young Scholars 11725106.

†Norwegian Research Center NORCE, Nygårdsgaten 112, 5008 Bergen, Norway (xtai@norceresearch.no).

‡Department of Mathematics, University of Oslo, 0316 Oslo, Norway (rwinther@math.uio.no).

§Corresponding author. Henan Academy of Big Data, Zhengzhou University, Zhengzhou 450052, People's Republic of China, and School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, People's Republic of China (zhangxiaodi@lsec.cc.ac.cn).

¶LSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, People's Republic of China, and School of Mathematical Science, University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China (zwy@lsec.cc.ac.cn).

where f is the observed image, $\alpha > 0$ is the penalization parameter which controls the trade-off between goodness-of-fit and visibility in its minimizer u, and $\mathrm{BV}(\Omega)$ denotes the space of functions of bounded variation [4]. The Euler–Lagrange equation for the minimization problem (1.1) can be formally written as

$$-\alpha\nabla\cdot\left(\frac{\nabla u}{|\nabla u|}\right)+u-f=0 \quad \text{in } \Omega, \qquad \nabla u\cdot\boldsymbol{n}=0 \quad \text{on } \partial\Omega. \tag{1.2}$$

Here and after, $\boldsymbol{n}$ denotes the unit outer normal vector of $\partial\Omega$. This equation indeed characterizes the first-order optimality condition of (1.1), which is also known as the curvature equation [36]. As a fundamental well-studied model in the literature, the ROF model is important for many modern applications, including scientific computing, image processing, and data sciences.

To deal with the singularity caused by the total variation seminorm minimization in (1.1), the following regularized minimization problem is often studied:

$$\min_{v\in \mathrm{W}^{1,1}(\Omega)}\left\{E_\beta(v)=\alpha\int_\Omega\sqrt{|\nabla v|^2+\beta}\mathrm{d}\boldsymbol{x}+\frac{1}{2}\int_\Omega(v-f)^2\mathrm{d}\boldsymbol{x}\right\}, \tag{1.3}$$

where the regularization parameter $\beta > 0$ is typically small and $\mathrm{W}^{1,1}(\Omega)$ is the Sobolev space of functions with absolute integrable function values and gradients. In [1], it has been shown that the solution of the regularized problem (1.3) converges to the solution of (1.1) as $\beta \to 0$. The convergence has been established rigorously in [49]. It should be noted that this regularization technique is mostly used to compute the minimizer of the total variation energy and its variants [12, 18].

It is well known that the first integral in (1.3) is the surface area of the graph of function v when $\beta = 1$. Thus, model (1.3) is essentially the minimum surface problem when $\beta = 1$. Since the regularized energy functional $E_\beta(v)$ is strictly convex for $\beta > 0$, the minimizer to (1.3) exists and is unique following classical analysis for minimum surface problems [35].

In this work, we will design a fast algorithm which has good convergence properties uniformly with respect to β, which means that our algorithm works for the regularized total variation minimization model as well as for the minimum surface model.

The Euler–Lagrange equation corresponding to the minimization problem (1.3) reads

$$-\alpha\nabla\cdot\left(\frac{\nabla u}{\sqrt{|\nabla u|^2+\beta}}\right)+u-f=0 \quad \text{in } \Omega, \qquad \nabla u\cdot\boldsymbol{n}=0 \quad \text{on } \partial\Omega. \tag{1.4}$$

We want to emphasize once more that our proposed method works uniformly with respect to $\beta \in (0, 1]$.

Numerical solution to the minimization problem (1.3) poses a challenging problem due to the presence of a highly nonlinear and nondifferentiable term. To get around this difficulty, a lot of effort has been devoted to constructing effective schemes for the minimization problem (1.3) and the PDE problems (1.2) in the past two decades, such as the artificial time marching scheme in [34, 37], the lagged diffusivity fixed-point method in [20, 44], the Chan–Golub–Mulet method in [17], the Bregman iteration in [22], the augmented Lagrangian technique in [46], the primal-dual approach in [24, 26], and some others in [3, 5, 6, 7, 13, 25, 30, 47]. However, most of these numerical algorithms are gradient-descent type and thus only first order. Thus, the main motivation of this work is to develop an effective second-order algorithm for the model (1.4).

It is well known that a "good" algorithm for the nonlinear problem should include not only fast iterative methods but also fast linear solvers for the linear systems obtained after linearization. Solving the linear systems is usually the most important, challenging, and time-consuming part in the overall simulation, which is due to the large-scale and ill conditions of the linear systems. For the underlying nonlinear PDE problem (1.4), the condition number of the linear system tends to infinity when the mesh size is approaching zero. Moreover, the variability of the parameters, such as the penalization parameter α and the regularization parameter β, can additionally lead to deterioration of the condition number. However, little work has been done to develop robust and efficient solvers for the resulting linear systems. Here, the term "robust" refers to the following properties: (i) The convergence rate is independent of the mesh size and other parameters of the model, and (ii) the number of iterations shall be independent of the mesh size and other model parameters. In this work, we achieve these goals through designing proper preconditioners for the linearized problem.

Based on the above discussion, the purpose of this paper is to propose a preconditioned Newton method for primal-dual finite element discretization of the ROF model. We shall adopt a primal-dual formulation of the model (1.4) which contains the primal variable u, the dual variable $\boldsymbol{p}$, and the multiplier $\boldsymbol{\lambda}$. We first discretize the primal-dual system using mixed finite element methods and then linearize the discrete nonlinear system by Newton's method. Following the operator preconditioning framework in [33], we develop block diagonal preconditioners for the linearized problem. The derivation exploits its well-posedness on appropriate Sobolev spaces equipped with proper norms. We also rigorously prove that the condition number of the preconditioned operator is uniformly bounded by a constant independent of the mesh size, the penalization parameter α, and the regularization parameter β. Thus, the proposed preconditioners are robust. Consequently, a second-order Newton scheme with robust and optimal preconditioners for the model (1.4) is given for the solution for our model. Finally, numerical experiments are supplied to test the accuracy of our schemes and validate that the proposed preconditioners are uniform with respect to mesh size, the penalization parameter α, and the regularization parameter β.

We want to emphasize that the essential ideas presented in this work are rather general and can be used for other nonlinear problems. In [27], the harmonic map problem was considered. A uniform preconditioner for the Newton iteration was also constructed using an operator preconditioning framework published later in [33]. All these show that the methodology presented here can be used for a large class of problems.

The remainder of the paper is structured as follows. In section 2, we introduce a primal-dual formulation of the ROF model and present its Newton iterations. Section 3 is devoted to introducing the finite element discretization and giving Newton's linearization for the discrete problem. In section 4, we derive the well-posedness of the discrete problem on chosen spaces equipped with subtle norms. We propose and analyze the robust preconditioners in section 5. We carry out several numerical experiments in section 6 to confirm the efficiency of our proposed algorithms. The paper ends with a concluding remark in section 7.

2. Our proposed model and the Newton algorithm. First, we introduce some Sobolev spaces and norms used in this paper. Throughout the paper, we shall denote vector-valued quantities by boldface notations. Let $L^2(\Omega)$ be the usual Hilbert space of square integrable functions which is equipped with the following inner product and norm:

$$\langle u, v\rangle := \int_\Omega u(\boldsymbol{x})v(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \qquad \|u\| \coloneqq \langle u, u\rangle^{1/2}.$$

Let $H^1(\Omega)$ be its subspace with square integrable gradients and its standard norm. We also use the space L^∞ with its canonical norm, $\|v\|_\infty = \operatorname{ess\,sup}_{\boldsymbol{x}\in\Omega}|v(\boldsymbol{x})|$. For a vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_d) \in \mathbb{R}^d$, we shall use the notation

$$|\boldsymbol{x}|_\beta = \sqrt{|\boldsymbol{x}|^2 + \beta} = \sqrt{\sum_{i=1}^{d} x_i^2 + \beta}.$$

For notation simplicity, we will also use $\langle\cdot,\cdot\rangle$ throughout this work to denote the L^2-type of inner product for vectors, functions, vector functions, and duality pairing. From the context in which this notation is used, it is clear which inner product or duality this notation is referring to. As usual, ∇ and $\nabla\cdot$ will be used as (distributional) gradient and divergence operators.

Inspired by [46], we introduce the auxiliary variable $\boldsymbol{p} = \nabla u$ and reformulate (1.3) into an equivalent constrained minimization problem:

$$\min_{\substack{\boldsymbol{p},u\\ \boldsymbol{p}=\nabla u}} \left\{ \int_\Omega \alpha|\boldsymbol{p}|_\beta \mathrm{d}\boldsymbol{x} + \frac{1}{2}\int_\Omega (u-f)^2 \mathrm{d}\boldsymbol{x} \right\}. \tag{2.1}$$

The constraint condition can be enforced by use of a Lagrange multiplier $\boldsymbol{\mu}$, and we then seek stationary points to the Lagrangian functional

$$\mathcal{L}(\boldsymbol{q}, v, \boldsymbol{\mu}) = \int_\Omega \left(\alpha|\boldsymbol{q}|_\beta + \frac{1}{2}(v-f)^2 - \boldsymbol{\mu}\cdot(\boldsymbol{q} - \nabla v) \right) \mathrm{d}\boldsymbol{x}.$$

Let $(\boldsymbol{p}, u, \boldsymbol{\lambda})$ be one saddle point for this problem. As in [29, 31, 40], the first-order optimality condition for (2.1) is

$$\alpha\boldsymbol{p}/|\boldsymbol{p}|_\beta - \boldsymbol{\lambda} = 0, \quad u - \nabla\cdot\boldsymbol{\lambda} = f, \quad -\boldsymbol{p} + \nabla u = 0 \quad \text{in } \Omega, \tag{2.2}$$

in conjunction with the following boundary conditions:

$$\boldsymbol{\lambda}\cdot\boldsymbol{n} = \nabla u\cdot\boldsymbol{n} = 0 \qquad \text{on } \partial\Omega. \tag{2.3}$$

Different from the original Euler–Lagrange equation (1.4) for u, this system contains three variables, i.e., functions u, $\boldsymbol{p}$, and $\boldsymbol{\lambda}$. Following [17], the method is dubbed the primal-dual method.

To solve the nonlinear system (2.2)–(2.3), we apply Newton's method as the linearization technique. For convenience, we write this system in the compact form $F(\boldsymbol{p}, u, \boldsymbol{\lambda}) = \boldsymbol{0}$, where F is the nonlinear map given by

$$F : (\boldsymbol{p}, u, \boldsymbol{\lambda}) \mapsto (\alpha\boldsymbol{p}/|\boldsymbol{p}|_\beta - \boldsymbol{\lambda}, u - f - \nabla\cdot\boldsymbol{\lambda}, -\boldsymbol{p} + \nabla u).$$

Therefore, Newton's method to solve this problem is as follows: Given $(\boldsymbol{p}^n, u^n, \boldsymbol{\lambda}^n)$, compute $\left(\boldsymbol{p}^{n+1}, u^{n+1}, \boldsymbol{\lambda}^{n+1}\right)$ by

$$\left(\boldsymbol{p}^{n+1}, u^{n+1}, \boldsymbol{\lambda}^{n+1}\right) = (\boldsymbol{p}^n, u^n, \boldsymbol{\lambda}^n) + (\delta\boldsymbol{p}^n, \delta u^n, \delta\boldsymbol{\lambda}^n), \tag{2.4}$$

where the correction $(\delta\boldsymbol{p}^n, \delta u^n, \delta\boldsymbol{\lambda}^n)$ is defined by

$$DF(\boldsymbol{p}^n, u^n, \boldsymbol{\lambda}^n)(\delta\boldsymbol{p}^n, \delta u^n, \delta\boldsymbol{\lambda}^n) = -F(\boldsymbol{p}^n, u^n, \boldsymbol{\lambda}^n). \tag{2.5}$$

As usual, $DF(\boldsymbol{p}^n, u^n, \boldsymbol{\lambda}^n)$ is the Fréchet derivative of the operator F at $(\boldsymbol{p}^n, u^n, \boldsymbol{\lambda}^n)$. For a given vector field $\boldsymbol{r}$, for simplicity, assume that $\boldsymbol{r}(\boldsymbol{x}) \neq 0$ for all $\boldsymbol{x} \in \Omega$, and define the matrix valued function $H(\boldsymbol{r}) = H(\boldsymbol{r}(\boldsymbol{x}))$ by

$$H(\boldsymbol{r}) = \frac{1}{|\boldsymbol{r}|_\beta}\left(\mathrm{I} - \frac{\boldsymbol{r}\boldsymbol{r}^t}{|\boldsymbol{r}|_\beta^2}\right).$$

Here $\boldsymbol{r}^t$ denotes the transpose of $\boldsymbol{r}$. We have that the matrix $H(\boldsymbol{r})$ is symmetric and satisfies

$$\frac{\beta}{|\boldsymbol{r}|_\beta^3} \leq \frac{H(\boldsymbol{r})\boldsymbol{\xi}\cdot\boldsymbol{\xi}}{\boldsymbol{\xi}\cdot\boldsymbol{\xi}} \leq \frac{1}{|\boldsymbol{r}|_\beta} \quad \forall \boldsymbol{\xi} \in \mathbb{R}^d \backslash \{\mathbf{0}\}. \tag{2.6}$$

The above estimate follows from the following observation:

$$\frac{|\boldsymbol{\xi}|^2|\beta}{|\boldsymbol{r}|_\beta^3} \leq H(\boldsymbol{r})\boldsymbol{\xi}\cdot\boldsymbol{\xi} = \frac{|\boldsymbol{\xi}|^2|\boldsymbol{r}|_\beta^2 - (\boldsymbol{\xi}\cdot\boldsymbol{r})^2}{|\boldsymbol{r}|_\beta^3} \leq \frac{|\boldsymbol{\xi}|^2}{|\boldsymbol{r}|_\beta}.$$

In addition, the matrix $H(\boldsymbol{r})$ is uniformly elliptic provided that $\boldsymbol{r} \in \boldsymbol{L}^\infty(\Omega)$. Namely, there exists a constant $c_0 > 0$ depending on $\|\boldsymbol{r}\|_{L^\infty}$ and β such that

$$H(\boldsymbol{r})\boldsymbol{\xi}\cdot\boldsymbol{\xi} \geq c_0|\boldsymbol{\xi}|^2 \quad \text{for a.e. } x \in \Omega \text{ and all } \boldsymbol{\xi} \in \mathbb{R}^d.$$

Let $g(t) = t/\sqrt{t^2+\beta}$, $t \in \mathbb{R}$. Then we have $g'(t) = 1/\sqrt{t^2+\beta} - t^2/(\sqrt{t^2+\beta})^3$. With this fact, direct calculation shows that the Fréchet derivative of the operator F at $(\boldsymbol{p}^n, u^n, \boldsymbol{\lambda}^n)$ reads

$$DF(\boldsymbol{p}^n, u^n, \boldsymbol{\lambda}^n) = \begin{pmatrix} \alpha H(\boldsymbol{p}^n) & 0 & -\mathbf{I} \\ 0 & \mathrm{I} & -\nabla\cdot \\ -\mathbf{I} & \nabla & 0 \end{pmatrix}.$$

As a consequence, for the nth step of the Newton iteration, the residual system (2.5) can be written as

$$\begin{cases} \alpha H(\boldsymbol{p}^n)\delta\boldsymbol{p}^n - \delta\boldsymbol{\lambda}^n & = \boldsymbol{r}_{\boldsymbol{p}}^n, \\ \delta u^n - \nabla\cdot\delta\boldsymbol{\lambda}^n & = r_u^n, \\ -\delta\boldsymbol{p}^n + \nabla\delta u^n & = \boldsymbol{r}_{\boldsymbol{\lambda}}^n, \end{cases} \tag{2.7}$$

where the right-hand sides are given by

$$\begin{aligned} \boldsymbol{r}_{\boldsymbol{p}}^n &\coloneqq -\alpha\boldsymbol{p}^n/|\boldsymbol{p}^n|_\beta + \boldsymbol{\lambda}^n, \\ r_u^n &\coloneqq f - u^n + \nabla\cdot\boldsymbol{\lambda}^n, \\ \boldsymbol{r}_{\boldsymbol{\lambda}}^n &\coloneqq \boldsymbol{p}^n - \nabla u^n. \end{aligned}$$

In subsequent sections, additional details on the iterations will be provided.

3. Finite element discretization. In this subsection, we introduce the finite element discretization of the system (2.2). Let $\mathcal{T}_h$ be a quasi-uniform and shape-regular simplex mesh of Ω with mesh size h. For any integer $k \geq 0, T \in \mathcal{T}_h$, let $P_k(T)$ be the space of polynomials of degree k, and define $\boldsymbol{P}_k(T) = (P_k(T))^d$. We associate a triple of piecewise polynomial, finite-dimensional spaces to approximate the solution $(\boldsymbol{p}, u, \boldsymbol{\lambda})$:

$$\begin{aligned} \boldsymbol{V}_h &\coloneqq \left\{\boldsymbol{q} \in \boldsymbol{L}^2(\Omega) : \boldsymbol{q}|_T \in \boldsymbol{P}_0(T) \quad \forall T \in \mathcal{T}_h\right\}, \\ U_h &\coloneqq \left\{v \in H^1(\Omega) : v|_T \in P_1(T) \quad \forall T \in \mathcal{T}_h\right\}, \\ \boldsymbol{W}_h &\coloneqq \left\{\boldsymbol{\mu} \in \boldsymbol{L}^2(\Omega) : \boldsymbol{\mu}|_T \in \boldsymbol{P}_0(T) \quad \forall T \in \mathcal{T}_h\right\}. \end{aligned}$$

Based on the above finite element spaces, the finite element approximation of the system (2.2) is formulated as follows: Find $(\boldsymbol{p}_h, u_h, \boldsymbol{\lambda}_h) \in \boldsymbol{V}_h \times U_h \times \boldsymbol{W}_h$ such that for any $(\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h) \in \boldsymbol{V}_h \times U_h \times \boldsymbol{W}_h$,

$$\begin{cases} \langle \alpha \boldsymbol{p}_h/|\boldsymbol{p}_h|_\beta, \boldsymbol{q}_h\rangle - \langle \boldsymbol{\lambda}_h, \boldsymbol{q}_h\rangle & = 0, \\ \langle u_h, v_h\rangle + \langle \boldsymbol{\lambda}_h, \nabla v_h\rangle & = \langle f, v_h\rangle, \\ -\langle \boldsymbol{p}_h, \boldsymbol{\mu}_h\rangle + \langle \nabla u_h, \boldsymbol{\mu}_h\rangle & = 0. \end{cases} \tag{3.1}$$

Since we are interested in developing fast solvers for the discrete problem, we do not elaborate on the well-posedness of (3.1) and simply assume that it has a unique solution.

We shall apply the Newton algorithm (2.5) to the above finite element system as well. Let $(\boldsymbol{p}_h^n, u_h^n, \boldsymbol{\lambda}_h^n) \in \boldsymbol{V}_h \times U_h \times \boldsymbol{W}_h$ be the approximate solutions of (3.1) at the nth Newton iteration. From the linearization in (2.7), for each step of the Newton iteration, the residual equation reads as follows: Find $(\delta\boldsymbol{p}_h^n, \delta u_h^n, \delta\boldsymbol{\lambda}_h^n) \in \boldsymbol{V}_h \times U_h \times \boldsymbol{W}_h$ such that for any $(\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h) \in \boldsymbol{V}_h \times U_h \times \boldsymbol{W}_h$,

$$\begin{cases} \langle \alpha H(\boldsymbol{p}_h^n)\delta\boldsymbol{p}_h^n, \boldsymbol{q}_h\rangle - \langle \delta\boldsymbol{\lambda}_h^n, \boldsymbol{q}_h\rangle & = R_{\boldsymbol{p}}^n(\boldsymbol{q}_h), \\ \langle \delta u_h^n, v_h\rangle + \langle \delta\boldsymbol{\lambda}_h^n, \nabla v_h\rangle & = R_u^n(v_h), \\ -\langle \delta\boldsymbol{p}_h^n, \boldsymbol{\mu}_h\rangle + \langle \nabla\delta u_h^n, \boldsymbol{\mu}_h\rangle & = R_{\boldsymbol{\lambda}}^n(\boldsymbol{\mu}_h), \end{cases} \tag{3.2}$$

where the residual functionals are defined by

$$\begin{aligned} R_{\boldsymbol{p}}^n(\boldsymbol{q}_h) &\coloneqq \langle -\alpha \boldsymbol{p}_h^n/|\boldsymbol{p}_h^n|_\beta + \boldsymbol{\lambda}_h^n, \boldsymbol{q}_h\rangle, \\ R_u^n(v_h) &\coloneqq \langle f - u_h^n, v_h\rangle - \langle \boldsymbol{\lambda}_h^n, \nabla v_h\rangle, \\ R_{\boldsymbol{\lambda}}^n(\boldsymbol{\mu}_h) &\coloneqq \langle \boldsymbol{p}_h^n - \nabla u_h^n, \boldsymbol{\mu}_h\rangle. \end{aligned}$$

Afterward, the new solution $\left(\boldsymbol{p}_h^{n+1}, u_h^{n+1}, \boldsymbol{\lambda}_h^{n+1}\right)$ is given by

$$\left(\boldsymbol{p}_h^{n+1}, u_h^{n+1}, \boldsymbol{\lambda}_h^{n+1}\right) = \left(\boldsymbol{p}_h^n, u_h^n, \boldsymbol{\lambda}_h^n\right) + \left(\delta\boldsymbol{p}_h^n, \delta u_h^n, \delta\boldsymbol{\lambda}_h^n\right). \tag{3.3}$$

Given a $\boldsymbol{r} \in \boldsymbol{V}_h$, we define the bilinear form $a_{\boldsymbol{r}}(\cdot,\cdot)$ as

$$\begin{aligned} a_{\boldsymbol{r}}\left((\boldsymbol{p}_h, u_h, \boldsymbol{\lambda}_h), (\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h)\right) \coloneqq\ & \langle \alpha H(\boldsymbol{r})\boldsymbol{p}_h, \boldsymbol{q}_h\rangle - \langle \boldsymbol{\lambda}_h, \boldsymbol{q}_h\rangle + \langle u_h, v_h\rangle \\ & + \langle \boldsymbol{\lambda}_h, \nabla v_h\rangle - \langle \boldsymbol{p}_h, \boldsymbol{\mu}_h\rangle + \langle \nabla u_h, \boldsymbol{\mu}_h\rangle \end{aligned}$$

and the residual functional $\ell_h(\cdot)$ as

$$\ell_h(\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h) \coloneqq R_{\boldsymbol{p}}^n(\boldsymbol{q}_h) + R_u^n(v_h) + R_{\boldsymbol{\lambda}}^n(\boldsymbol{\mu}_h).$$

Write $X_h \coloneqq \boldsymbol{V}_h \times U_h \times \boldsymbol{W}_h$. Then the problem (3.2) can be equivalently written as follows: Given $(\boldsymbol{p}_h^n, u_h^n, \boldsymbol{\lambda}_h^n) \in X_h$, find $(\delta\boldsymbol{p}_h^n, \delta u_h^n, \delta\boldsymbol{\lambda}_h^n) \in X_h$ such that

$$a_{\boldsymbol{p}_h^n}\left((\delta\boldsymbol{p}_h^n, \delta u_h^n, \delta\boldsymbol{\lambda}_h^n), (\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h)\right) = \ell_h(\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h) \quad \forall (\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h) \in X_h. \tag{3.4}$$

For a given $\boldsymbol{r} \in \boldsymbol{V}_h$, we introduce the operator $\mathcal{A}_h(\boldsymbol{r}) : X_h \to X_h^*$ defined by

$$\begin{aligned} \langle \mathcal{A}_h(\boldsymbol{r})(\boldsymbol{p}_h, u_h, \boldsymbol{\lambda}_h), (\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h)\rangle \coloneqq\ & a_{\boldsymbol{r}}\left((\boldsymbol{p}_h, u_h, \boldsymbol{\lambda}_h), (\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h)\right), \\ & \forall (\boldsymbol{p}_h^n, u_h^n, \boldsymbol{\lambda}_h^n),\ (\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h) \in X_h. \end{aligned} \tag{3.5}$$

We remind the reader that $\langle \cdot, \cdot \rangle$ here refers to the duality pairing between X_h^* and X_h. Roughly speaking, $\mathcal{A}_h(\boldsymbol{r})$ is the discrete version of the following operator:

$$\mathcal{A}(\boldsymbol{r}) \coloneqq \begin{pmatrix} \alpha H(\boldsymbol{r}) & 0 & -\mathbf{I} \\ 0 & \mathrm{I} & -\nabla\cdot \\ -\mathbf{I} & \nabla & 0 \end{pmatrix}. \tag{3.6}$$

We first note that the form $\langle \mathcal{A}_h(\boldsymbol{r})\cdot, \cdot \rangle$ is symmetric, reflecting the symmetry of the saddle point operator $\mathcal{A}_h(\boldsymbol{r})$. Besides that, for the nth step of the Newton iteration, we need to solve a linear system with the operator $\mathcal{A}_h(\boldsymbol{p}_h^n)$ as the coefficient matrix. Since the operator $\mathcal{A}_h(\boldsymbol{r})$ is indefinite and symmetric, we solve the linear system by the minimum residual method (MINRES) [33]. However, as one of the Krylov space methods, the convergence rate of MINRES depends on the condition number and the spectrum of $\mathcal{A}_h(\boldsymbol{r})$. This motivates the study of efficient preconditioners for the operator $\mathcal{A}_h(\boldsymbol{r})$.

Remark 3.1. From [8], using the inverse estimate in the finite element space $\boldsymbol{V}_h$, we have

$$\|\boldsymbol{p}_h\|_\infty \leq C h^{-d/2} \|\boldsymbol{p}_h\|. \tag{3.7}$$

Thus, given a mesh size $h > 0$, $H(\boldsymbol{p}_h^n) \in \boldsymbol{L}^\infty(\Omega)$ is uniformly elliptic; see (2.6). Accordingly, problems (3.1) and (3.2) are well-defined.

4. Well-posedness of the linear system for the Newton algorithm. Now we discuss the well-posedness of scheme (3.4), which is the foundation of the preconditioners we proposed. For this end, we will use the Babuška–Brezzi theory [2, 9, 10] to analyze the mapping properties of the operator $\mathcal{A}_h(\boldsymbol{r})$ for any $\boldsymbol{r} \in \boldsymbol{V}_h$.

As discussed in [33], ensuring that the continuity constants and the inf-sup constants are independent of the physical parameters and the discretized parameters is crucial to design robust block preconditioners for solving our problem. Note that the natural bound on the saddle point operator $\mathcal{A}_h(\boldsymbol{r})$ depends on the vector field $\boldsymbol{r}$. Therefore, we equip the space X_h with the $\boldsymbol{r}$-dependent inner product

$$\begin{aligned} ((\boldsymbol{p}_h, u_h, \boldsymbol{\lambda}_h), (\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h))_{X_{\boldsymbol{r},\alpha}} \coloneqq & \alpha \langle H(\boldsymbol{r})\boldsymbol{p}_h, \boldsymbol{q}_h \rangle + \langle u_h, v_h \rangle + \alpha \langle H(\boldsymbol{r}) \nabla u_h, \nabla v_h \rangle \\ & + \alpha^{-1} \langle H(\boldsymbol{r})^{-1} \boldsymbol{\lambda}_h, \boldsymbol{\mu}_h \rangle \end{aligned} \tag{4.1}$$

and norm

$$\begin{aligned} \|(\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h)\|_{X_{\boldsymbol{r},\alpha}}^2 \coloneqq & \alpha \langle H(\boldsymbol{r})\boldsymbol{q}_h, \boldsymbol{q}_h \rangle + \|v_h\|^2 + \alpha \langle H(\boldsymbol{r}) \nabla v_h, \nabla v_h \rangle \\ & + \alpha^{-1} \langle H(\boldsymbol{r})^{-1} \boldsymbol{\mu}_h, \boldsymbol{\mu}_h \rangle. \end{aligned} \tag{4.2}$$

It is clear that the corresponding norm for the dual space X_h^* depends on $\boldsymbol{r}$ and α. Thanks to the estimate (3.7) and (2.6), the matrix $H(\boldsymbol{r})$ is uniformly elliptic and invertible. Hence, the weighted norm (4.2) is well-defined in X_h.

After the introduction of these notations, we are able to show that the operator $\mathcal{A}_h(\boldsymbol{r})$ has the following properties.

LEMMA 4.1 (boundedness of $\mathcal{A}_h(\boldsymbol{r})$). *For all* $(\boldsymbol{p}_h, u_h, \boldsymbol{\lambda}_h)$, $(\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h) \in X_h$, *we have*

$$|\langle \mathcal{A}_h(\boldsymbol{r})(\boldsymbol{p}_h, u_h, \boldsymbol{\lambda}_h), (\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h) \rangle| \leq 2 \|(\boldsymbol{p}_h, u_h, \boldsymbol{\lambda}_h)\|_{X_{\boldsymbol{r},\alpha}} \|(\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h)\|_{X_{\boldsymbol{r},\alpha}}.$$

Proof. This follows directly from the definition of the operator $\mathcal{A}_h(\boldsymbol{r})$ and the definition of $\|\cdot\|_{X_r}$. ∎

We define the associated kernel space $Z_h \subset \boldsymbol{V}_h \times U_h$ by

$$Z_h = \{(\boldsymbol{p}_h, u_h) \in \boldsymbol{V}_h \times U_h \mid \langle \nabla u_h - \boldsymbol{p}_h, \boldsymbol{\mu}_h \rangle = 0 \quad \forall \boldsymbol{\mu}_h \in \boldsymbol{W}_h\}.$$

Note that $\nabla u_h \in \boldsymbol{W}_h$ and $\boldsymbol{p}_h \in \boldsymbol{W}_h$; thus, taking $\boldsymbol{\mu}_h = \nabla u_h - \boldsymbol{p}_h$ gives $\boldsymbol{p}_h = \nabla u_h$. Consequently, the discrete kernel space Z_h is further portrayed as

$$Z_h = \{(\boldsymbol{p}_h, u_h) \in \boldsymbol{V}_h \times U_h \mid \boldsymbol{p}_h = \nabla u_h\}.$$

LEMMA 4.2 (coercivity on the kernel space). *For all $(\boldsymbol{p}_h, u_h) \in Z_h$, we have*

$$\langle \mathcal{A}_h(\boldsymbol{r})(\boldsymbol{p}_h, u_h, 0), (\boldsymbol{p}_h, u_h, 0)\rangle \geq \frac{1}{2} \|(\boldsymbol{p}_h, u_h, 0)\|^2_{X_{\boldsymbol{r},\alpha}}.$$

Proof. Just observe that on Z_h

$$\begin{aligned}\langle \mathcal{A}_h(\boldsymbol{r})(\boldsymbol{p}_h, u_h, 0), (\boldsymbol{p}_h, u_h, 0)\rangle &= \langle \alpha H(\boldsymbol{r})\boldsymbol{p}_h, \boldsymbol{p}_h\rangle + \|u_h\|^2 \\ &= \frac{1}{2}\langle \alpha H(\boldsymbol{r})\boldsymbol{p}_h, \boldsymbol{p}_h\rangle + \|u_h\|^2 + \frac{1}{2}\langle \alpha H(\boldsymbol{r})\nabla u_h, \nabla u_h\rangle\end{aligned}$$

and that

$$\|(\boldsymbol{p}_h, u_h, 0)\|^2_{X_{\boldsymbol{r},\alpha}} = \langle \alpha H(\boldsymbol{r})\boldsymbol{p}_h, \boldsymbol{p}_h\rangle + \|u_h\|^2 + \alpha\langle H(\boldsymbol{r})\nabla u_h, \nabla u_h\rangle.$$

Then the desired inequality holds. ∎

LEMMA 4.3 (inf-sup condition). *The following estimate holds:*

$$\sup_{(\boldsymbol{q}_h, v_h) \in \boldsymbol{V}_h \times U_h} \frac{\langle \mathcal{A}_h(\boldsymbol{r})(0,0,\boldsymbol{\lambda}_h), (\boldsymbol{q}_h, v_h, 0)\rangle}{\|(\boldsymbol{q}_h, v_h, 0)\|_{X_{\boldsymbol{r},\alpha}}} \geq \|(0,0,\boldsymbol{\lambda}_h)\|_{X_{\boldsymbol{r},\alpha}} \quad \forall \boldsymbol{\lambda} \in \boldsymbol{W}_h.$$

Proof. We have that

$$\begin{aligned}\sup_{(\boldsymbol{q}_h, v_h) \in \boldsymbol{V}_h \times U_h} \frac{\langle \mathcal{A}_h(\boldsymbol{r})(0,0,\boldsymbol{\lambda}_h), (\boldsymbol{q}_h, v_h, 0)\rangle}{\|(\boldsymbol{q}_h, v_h, 0)\|_{X_{\boldsymbol{r},\alpha}}} &\geq \sup_{(\boldsymbol{q}_h, 0) \in \boldsymbol{V}_h \times U_h} \frac{-\langle \boldsymbol{\lambda}_h, \boldsymbol{q}_h\rangle}{(\alpha\langle H(\boldsymbol{r})\boldsymbol{q}_h, \boldsymbol{q}_h\rangle)^{1/2}} \\ &\geq \left(\alpha^{-1}\langle H(\boldsymbol{r})^{-1}\boldsymbol{\lambda}_h, \boldsymbol{\lambda}_h\rangle\right)^{1/2},\end{aligned}$$

where the last inequality follows by taking $\boldsymbol{q}_h = -\alpha^{-1}H(\boldsymbol{r})^{-1}\boldsymbol{\lambda}_h$. Since

$$\|(0,0,\boldsymbol{\lambda}_h)\|_{X_{\boldsymbol{r},\alpha}} = \left(\alpha^{-1}\langle H(\boldsymbol{r})^{-1}\boldsymbol{\lambda}_h, \boldsymbol{\lambda}_h\rangle\right)^{1/2},$$

this shows that the desired inequality holds. ∎

With the three lemmas above, we obtain the main result of this section.

THEOREM 4.1. *At each iteration for the Newton updating, the discretized problem* (3.2) *is well-posed.*

Proof. In Lemmas 4.1–4.3, we have verified Brezzi conditions for the saddle point problem. Therefore, it is straightforward to reach the conclusion by using the Babuška–Brezzi theory. ∎

Using adequately weighted spaces, we have that $\mathcal{A}_h(\boldsymbol{r})$ is an isomorphism from X_h to X_h^* such that $\|\mathcal{A}_h(\boldsymbol{r})\|_{\mathcal{L}(X_h, X_h^*)}$ and $\|(\mathcal{A}_h(\boldsymbol{r}))^{-1}\|_{\mathcal{L}(X_h^*, X_h)}$ are bounded independently of mesh sizes and the parameters. That is,

$$\|\mathcal{A}_h(\boldsymbol{r})\|_{\mathcal{L}(X_h, X_h^*)} \leq C, \quad \left\|(\mathcal{A}_h(\boldsymbol{r}))^{-1}\right\|_{\mathcal{L}(X_h^*, X_h)} \leq c^{-1}, \tag{4.3}$$

where the constants C and c are independent of $\boldsymbol{r}, \alpha, \beta$, and h.

5. Robust preconditioners. In this section, we develop and analyze robust preconditioners. Let $\mathcal{B}_h(\boldsymbol{r})$ be the "Riesz-operator" mapping from X_h^* to X_h, which is induced by the weighted norm $\|\cdot\|_{X_{\boldsymbol{r},\alpha}}$ for given $(\boldsymbol{p}_h, u_h, \boldsymbol{\lambda}_h) \in X_h^*$:

$$(\mathcal{B}_h(\boldsymbol{r})(\boldsymbol{p}_h, u_h, \boldsymbol{\lambda}_h), (\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h))_{X_{\boldsymbol{r},\alpha}} \coloneqq \langle (\boldsymbol{p}_h, u_h, \boldsymbol{\lambda}_h), (\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h) \rangle, \quad (\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h) \in X_h.$$

By using (4.1) and (4.2), we can see that it takes the following explicit form:

$$\mathcal{B}_h(\boldsymbol{r}) = \begin{pmatrix} \alpha H(\boldsymbol{r}) & 0 & 0 \\ 0 & S_{\boldsymbol{r},h} & 0 \\ 0 & 0 & \alpha^{-1} H(\boldsymbol{r})^{-1} \end{pmatrix}^{-1}, \tag{5.1}$$

where the operator $S_{\boldsymbol{r},h} \colon V_h \mapsto V_h^*$ is defined by

$$\langle S_{\boldsymbol{r},h} u_h, v_h \rangle \coloneqq \langle u_h, v_h \rangle + \langle \alpha H(\boldsymbol{r}) \nabla u_h, \nabla v_h \rangle \qquad \forall u_h, v_h \in V_h.$$

In fact, $S_{\boldsymbol{r},h}$ is the finite element discretization for the operator $\mathrm{I} - \alpha \nabla \cdot (H(\boldsymbol{r}) \nabla)$ with Neumann boundary conditions. Note that since the matrix $H(\boldsymbol{r})$ is symmetric and uniformly elliptic, the second block, $S_{\boldsymbol{r},h}$, is invertible. Following the operator preconditioning framework [33], the operator $\mathcal{B}_h(\boldsymbol{r})$ is proposed as the preconditioner for $\mathcal{A}_h(\boldsymbol{r})$.

In the following, we estimate the condition number of the preconditioned operator $\mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})$ using the formula

$$\kappa\left(\mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\right) = \|\mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\|_{\mathcal{L}(X_h, X_h)} \left\| \left(\mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\right)^{-1} \right\|_{\mathcal{L}(X_h, X_h)}. \tag{5.2}$$

THEOREM 5.1. *The condition number $\kappa\left(\mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\right)$ has a uniform bound, independent of $\boldsymbol{r}, \alpha, \beta$, and h, in the sense that*

$$1 \le \kappa\left(\mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\right) \le C/c,$$

where C and c are defined by (4.3).

Proof. From the definition of the operator $\mathcal{B}_h(\boldsymbol{r})$ and the definition of $\|\cdot\|_{X_{\boldsymbol{r},\alpha}}$, we note that the operator $\mathcal{B}_h(\boldsymbol{r})$ has the property that

$$\|\mathcal{B}_h(\boldsymbol{r})\|_{\mathcal{L}(X_h^*, X_h)} = \left\| \left(\mathcal{B}_h(\boldsymbol{r})\right)^{-1} \right\|_{\mathcal{L}(X_h, X_h^*)} = 1.$$

Using (4.3), we further have

$$\|\mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\|_{\mathcal{L}(X_h, X_h)} = \|\mathcal{A}_h(\boldsymbol{r})\|_{\mathcal{L}(X_h, X_h^*)} \le C$$

and

$$\left\| \left(\mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\right)^{-1} \right\|_{\mathcal{L}(X_h, X_h)} \le \left\| \left(\mathcal{A}_h(\boldsymbol{r})\right)^{-1} \right\|_{\mathcal{L}(X_h^*, X_h)} \left\| \left(\mathcal{B}_h(\boldsymbol{r})\right)^{-1} \right\|_{\mathcal{L}(X_h, X_h^*)} \le c^{-1}.$$

Finally, we get the desired result using (5.2). ∎

This theorem suggests that $\mathcal{B}_h(\boldsymbol{r})$ is a "good" preconditioner for $\mathcal{A}_h(\boldsymbol{r})$ in the sense that the preconditioned operator has a condition number which is independent of the mesh size h, the penalization parameter α, the regularization parameter β, and the iterative step n. We are using MINRES to solve the preconditioned linear system at each Newton iteration. We have the following convergence result for MINRES.

THEOREM 5.2. *If x^0 is the initial value, x^m is the mth iteration of the MINRES method, and x is the exact solution, then there exists a constant $\delta \in (0,1)$, only depending on the condition number $\kappa(\mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r}))$, such that*

$$\begin{aligned}&\langle \mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\left(x-x^m\right), \mathcal{A}_h(\boldsymbol{r})\left(x-x^m\right)\rangle^{\frac{1}{2}}\\ &\quad\le 2\delta^m \left\langle \mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\left(x-x^0\right), \mathcal{A}_h(\boldsymbol{r})\left(x-x^0\right)\right\rangle^{\frac{1}{2}}.\end{aligned}$$

Moreover, an estimate leads to

$$\delta = \frac{\kappa(\mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r}))-1}{\kappa(\mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r}))+1} \le \frac{C-c}{C+c}.$$

Proof. Since the operator $\mathcal{B}_h(\boldsymbol{r}) \in \mathcal{L}(X_h^*, X_h)$ is symmetric and positive definite, $\langle (\mathcal{B}_h(\boldsymbol{r}))^{-1}\cdot,\cdot\rangle$ defines an inner product on X_h. Furthermore, it is easy to see that the preconditioned operator $\mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r}) \in \mathcal{L}(X_h, X_h)$ is symmetric in this inner product. So we shall use the MINRES method, which is defined with respect to the inner product $\langle (\mathcal{B}_h(\boldsymbol{r}))^{-1}\cdot,\cdot\rangle$. The estimate is obtained by applying Theorem 2.2 in [33]. ∎

The inverse of $S_{\boldsymbol{r},h}$ needs to solve an elliptic linear problem. For large-scale computations, it is prohibitively expensive and time consuming to use direct methods to solve this elliptic problem, especially for three-dimensional problems. Thus, we consider the following inexact preconditioner:

$$\widetilde{\mathcal{B}}_h(\boldsymbol{r}) = \begin{pmatrix} \alpha H(\boldsymbol{r}) & 0 & 0 \\ 0 & \widetilde{S}_{\boldsymbol{r},h}^{-1} & 0 \\ 0 & 0 & \alpha^{-1}H(\boldsymbol{r})^{-1} \end{pmatrix}^{-1}. \tag{5.3}$$

Here, $\widetilde{S}_{\boldsymbol{r},h}$ is an operator that is spectrally equivalent to the action of the inverse of the block $S_{\boldsymbol{r},h}$ in the following sense:

$$c_{1,s}\langle S_{\boldsymbol{r},h}^{-1}v_h, v_h\rangle \le \langle \widetilde{S}_{\boldsymbol{r},h}v_h, v_h\rangle \le c_{2,s}\langle S_{\boldsymbol{r},h}^{-1}v_h, v_h\rangle \quad \forall v_h \in V_h, \tag{5.4}$$

where the constants $c_{1,s}$ and $c_{2,s}$ are independent of the mesh size, the penalization parameter, the regularization parameter, and the iterative step. Recall that $S_{\boldsymbol{r},h}$ is the finite element discretization of the operator $\mathrm{I}-\alpha\nabla\cdot(H(\boldsymbol{r})\nabla)$ with Neumann boundary conditions. There are many methods to construct efficient preconditioners for such a type of operator, such as the standard multigrid method [11, 23, 38], the algebraic multigrid method (AMG) [32, 43, 48], and the domain decomposition method [21, 42]. When $H(\boldsymbol{r})$ is the identity matrix and $\alpha > 0$ (can be small), most of the mentioned methods can guarantee that the operator $\tilde{S}_{\boldsymbol{r},h}$ satisfies the requirement (5.4) uniformly with respect to α; see [33, section 6] and references given there. However, we have not made any attempt to establish a similar theoretical result for the more general matrix $H(\boldsymbol{r})$ used here since we regard this to be out of the main scope of the present discussion. Instead, we make the assumption that (5.4) holds, and our experiments seem to confirm that this is indeed the case.

When using $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$ as the preconditioner, we have the following results regarding the convergence estimate of the corresponding preconditioned MINRES method.

THEOREM 5.3. *Assume that* (5.4) *is satisfied. Then we have*

$$1 \le \kappa\left(\widetilde{\mathcal{B}}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\right) \le \frac{C\hat{c}_2}{c\hat{c}_1},$$

where $\hat{c}_1 = \min(c_{1,s}, 1)$ *and* $\hat{c}_2 = \max(c_{2,s}, 1)$. *Moreover, if* x^0 *is the initial value,* x^m *is the* m*th iteration of the MINRES method, and* x *is the exact solution, then there exists a constant* $\delta \in (0,1)$*, only depending on the condition number* $\kappa(\widetilde{\mathcal{B}}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r}))$*, such that*

$$\left\langle \widetilde{\mathcal{B}}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\left(x - x^m\right), \mathcal{A}_h(\boldsymbol{r})\left(x - x^m\right)\right\rangle^{\frac{1}{2}} \leq 2\delta^m \left\langle \widetilde{\mathcal{B}}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\left(x - x^0\right), \mathcal{A}_h(\boldsymbol{r})\left(x - x^0\right)\right\rangle^{\frac{1}{2}}.$$

Furthermore, an estimate leads to

$$\delta = \frac{\kappa\left(\widetilde{\mathcal{B}}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\right) - 1}{\kappa\left(\widetilde{\mathcal{B}}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\right) + 1} \leq \frac{\hat{c}_2 C - \hat{c}_1 c}{\hat{c}_2 C + \hat{c}_1 c}.$$

Proof. We only need to estimate the condition number of the preconditioned operator $\widetilde{\mathcal{B}}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})$; the result of the convergence of the preconditioned MINRES method is obvious. From (5.4), we can see that $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$ satisfies

$$\hat{c}_1 (x,x)_{\mathcal{B}_h(\boldsymbol{r})} \leq (x,x)_{\widetilde{\mathcal{B}}_h(\boldsymbol{r})} \leq \hat{c}_2 (x,x)_{\mathcal{B}_h(\boldsymbol{r})}. \tag{5.5}$$

This implies that

$$\left\|\widetilde{\mathcal{B}}_h(\boldsymbol{r})\left(\mathcal{B}_h(\boldsymbol{r})\right)^{-1}\right\|_{\mathcal{L}(X_h,X_h)} \leq \hat{c}_2, \quad \left\|\left(\widetilde{\mathcal{B}}_h(\boldsymbol{r})\left(\mathcal{B}_h(\boldsymbol{r})\right)^{-1}\right)^{-1}\right\|_{\mathcal{L}(X_h,X_h)} \leq \hat{c}_1^{-1}.$$

Similar to the proof of Theorem 5.1, we have

$$\begin{aligned}\left\|\widetilde{\mathcal{B}}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\right\|_{\mathcal{L}(X_h,X_h)} &= \left\|\widetilde{\mathcal{B}}_h(\boldsymbol{r})\left(\mathcal{B}_h(\boldsymbol{r})\right)^{-1}\mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\right\|_{\mathcal{L}(X_h,X_h)} \\ &\leq \left\|\widetilde{\mathcal{B}}_h(\boldsymbol{r})\left(\mathcal{B}_h(\boldsymbol{r})\right)^{-1}\right\|_{\mathcal{L}(X_h,X_h)} \|\mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\|_{\mathcal{L}(X_h,X_h)} \leq C\hat{c}_2\end{aligned}$$

and

$$\begin{aligned}\left\|\left(\widetilde{\mathcal{B}}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\right)^{-1}\right\|_{\mathcal{L}(X_h,X_h)} &= \left\|\left(\widetilde{\mathcal{B}}_h(\boldsymbol{r})\left(\mathcal{B}_h(\boldsymbol{r})\right)^{-1}\mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\right)^{-1}\right\|_{\mathcal{L}(X_h,X_h)} \\ &\leq \left\|\left(\mathcal{B}_h(\boldsymbol{r})\mathcal{A}_h(\boldsymbol{r})\right)^{-1}\right\|_{\mathcal{L}(X_h,X_h)} \left\|\left(\widetilde{\mathcal{B}}_h(\boldsymbol{r})\left(\mathcal{B}_h(\boldsymbol{r})\right)^{-1}\right)^{-1}\right\|_{\mathcal{L}(X_h,X_h)} \leq c^{-1}\hat{c}_1^{-1}.\end{aligned}$$

Therefore, we get the estimate of the condition number. ∎

As a consequence, the exact methods for the block solvers in the preconditioner $\mathcal{B}_h(\boldsymbol{r})$ can be replaced by inexact methods and still maintain the desired properties. So far, we have gotten a robust and effective solver when solving the linearized systems. This makes it very cheap to solve the system for each iteration for Newton updating. Together, it yields a second-order Newton scheme with robust and optimal preconditioners.

6. Numerical experiments. In this section, we present some numerical experiments to verify the convergence rate of the finite element approximation to our model and to demonstrate the robustness of the preconditioners. A set of two-dimensional examples are reported below. All codes were written in MATLAB based on the open-source finite element library *i*FEM [19].

For comparison, we also implement the following fixed point method, which is also known as the Picard method. For this method, given the nth iteration $(\boldsymbol{p}_h^n, u_h^n, \boldsymbol{\lambda}_h^n) \in \boldsymbol{V}_h \times U_h \times \boldsymbol{W}_h$, $\left(\boldsymbol{p}_h^{n+1}, u_h^{n+1}, \boldsymbol{\lambda}_h^{n+1}\right) \in \boldsymbol{V}_h \times U_h \times \boldsymbol{W}_h$ is solved by

$$\begin{cases} \langle \alpha \boldsymbol{p}_h^{n+1} / |\boldsymbol{p}_h^n|_\beta, \boldsymbol{q}_h \rangle - \langle \boldsymbol{\lambda}_h^{n+1}, \boldsymbol{q}_h \rangle = 0 & \forall \boldsymbol{q}_h \in \boldsymbol{V}_h, \\ \langle u_h^{n+1}, v_h \rangle + \langle \boldsymbol{\lambda}_h^{n+1}, \nabla v_h \rangle - \langle f, v_h \rangle = 0 & \forall v_h \in U_h, \\ -\langle \boldsymbol{p}_h^{n+1}, \boldsymbol{\mu}_h \rangle + \langle \nabla u_h^{n+1}, \boldsymbol{\mu}_h \rangle = 0 & \forall \boldsymbol{\mu}_h \in \boldsymbol{W}_h. \end{cases} \tag{6.1}$$

With the notation in section 3, its residual form reads as follows: Find $(\delta \boldsymbol{p}_h^n, \delta u_h^n, \delta \boldsymbol{\lambda}_h^n) \in \boldsymbol{V}_h \times U_h \times \boldsymbol{W}_h$ such that for any $(\boldsymbol{q}_h, v_h, \boldsymbol{\mu}_h) \in \boldsymbol{V}_h \times U_h \times \boldsymbol{W}_h$,

$$\begin{cases} \langle \alpha \widehat{H}(\boldsymbol{p}_h^n) \delta \boldsymbol{p}_h^n, \boldsymbol{q}_h \rangle - \langle \delta \boldsymbol{\lambda}_h^n, \boldsymbol{q}_h \rangle = R_{\boldsymbol{p}}^n(\boldsymbol{q}_h), \\ \langle \delta u_h^n, v_h \rangle + \langle \delta \boldsymbol{\lambda}_h^n, \nabla v_h \rangle = R_u^n(v_h), \\ -\langle \delta \boldsymbol{p}_h^n, \boldsymbol{\mu}_h \rangle + \langle \nabla \delta u_h^n, \boldsymbol{\mu}_h \rangle = R_{\boldsymbol{\lambda}}^n(\boldsymbol{\mu}_h), \end{cases} \tag{6.2}$$

where $\widehat{H}(\boldsymbol{p}_h^n) \coloneqq 1/|\boldsymbol{p}_h^n|_\beta$ and the residuals $R_{\boldsymbol{p}}^n(\boldsymbol{q}_h)$, R_u^n, and $R_{\boldsymbol{\lambda}}^n$ are given in section 3. Thus, the new solution $\left(\boldsymbol{p}_h^{n+1}, u_h^{n+1}, \boldsymbol{\lambda}_h^{n+1}\right)$ is given by

$$\left(\boldsymbol{p}_h^{n+1}, u_h^{n+1}, \boldsymbol{\lambda}_h^{n+1}\right) = (\boldsymbol{p}_h^n, u_h^n, \boldsymbol{\lambda}_h^n) + (\delta \boldsymbol{p}_h^n, \delta u_h^n, \delta \boldsymbol{\lambda}_h^n). \tag{6.3}$$

Clearly, the difference of the Newton method and the Picard method lies on the matrix $H(\boldsymbol{r})$ and the scalar $\widehat{H}(\boldsymbol{r})$. Note that $\widehat{H}(\boldsymbol{r})$ is bounded above by $1/\beta$ and below by zero. Hence, the theory for the Newton method in sections 4–5 also applies to the Picard method as long as we replace $H(\boldsymbol{r})$ by $\widehat{H}(\boldsymbol{r})$ in $\mathcal{A}_h(\boldsymbol{r})$, $\mathcal{B}_h(\boldsymbol{r})$, and $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$. For convenience, we still use the notation $\mathcal{A}_h(\boldsymbol{r})$, $\mathcal{B}_h(\boldsymbol{r})$, and $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$ for the Picard method. From (3.2), (6.2), (5.1), and (5.3), we know that the costs per iteration for the Picard method and the Newton method are almost the same. Thus, for convenience, only the iteration numbers needed by both methods are used for comparison in this section.

6.1. Implementation of block preconditioners. First of all, we discuss some implementation details of the proposed block preconditioners. To solve the linear system obtained from the finite element discretization, we use the MINRES method as an outer iterative solver, with the tolerance for the relative residual in the energy norm set to $\varepsilon = 10^{-10}$. The block preconditioners designed in section 5 are used to accelerate the convergence rate of MINRES. In the following, we implement both exact and inexact inner solvers, $\mathcal{B}_h(\boldsymbol{r})$ and $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$. As we know, inverting the proposed block preconditioners ends up with inverting diagonal blocks. Therefore, the main difference in implementation is how to invert the second diagonal block. For the exact preconditioner $\mathcal{B}_h(\boldsymbol{r})$, we call direct solvers implemented in MATLAB, while for the inexact preconditioner $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$, we mainly call the AMG preconditioned conjugate gradient (PCG) method to define the operator $\widetilde{S}_{\boldsymbol{r},h}$. The tolerance of PCG in terms of the l_2-norm of the relative residual is $\varepsilon_0 = 10^{-3}$. The AMG we used is the classical AGM with Ruge–Stuben coarsening and standard interpolation [19, 48].

Without specifications, the initial guess $\left(\boldsymbol{p}_h^0, u_h^0, \boldsymbol{\lambda}_h^0\right)$ is taken to be the zero solution, and the relative tolerances are set by 10^{-6} for the nonlinear iteration. Here the maximal iteration number of the MINRES solver is set by $N = 200$. For the sake of convenience, we denote N_{Newton} by the number of Newton iterations, N_{Picard} by the number of Picard iterations, and N_{MINRES} by the average number of preconditioned MINRES iterations for solving the linearized problem.

To ensure the global convergence of the Newton algorithm, we introduce an additional damping parameter θ. Let $\boldsymbol{b}^n$ be the vector representation of the right hand of (3.4) at nth iteration. We use the classical backtracking line search method, which selects step length θ^n to be the first number in the sequence of $\left\{1/2^k\right\}_{k=0}^{\infty}$ that satisfies the following criterion:

$$\left\|\boldsymbol{b}^{k+1}\right\|_{l^2} \leq (1-\sigma\theta_k)\left\|\boldsymbol{b}^n\right\|_{l^2},$$

where σ is chosen as 10^{-4}. Thus, the damped Newton updating is given by

$$\left(\boldsymbol{p}_h^{n+1}, u_h^{n+1}, \boldsymbol{\lambda}_h^{n+1}\right) = \left(\boldsymbol{p}_h^n, u_h^n, \boldsymbol{\lambda}_h^n\right) + \theta^n\left(\delta\boldsymbol{p}_h^n, \delta u_h^n, \delta\boldsymbol{\lambda}_h^n\right). \tag{6.4}$$

6.2. Numerical results. This subsection reports on some numerical experiments.

Example 6.1. This example is to test the convergence rate of finite element solutions and the robustness of the preconditioners for some smooth problems. The computational domain Ω is set as $(0,1)^2$. The function f is chosen so that the exact solutions are given by

$$\begin{aligned} u &= \cos(\pi x)\cos(\pi y), \\ \boldsymbol{p} &= -\pi\left(\sin(\pi x)\cos(\pi y), \cos(\pi x)\sin(\pi y)\right)^{\top}, \\ \boldsymbol{\lambda} &= -\frac{\pi\alpha}{|\boldsymbol{p}|_\beta}\left(\sin(\pi x)\cos(\pi y), \cos(\pi x)\sin(\pi y)\right)^{\top}. \end{aligned}$$

We first carry out the numerical experiment with $\alpha = \beta = 1$. The computational meshes are a set of uniform triangulations of Ω. Table 6.1 shows the information for the meshes and the number of degrees of freedom (DOFs) which we are using. From the finite element spaces in section 3, the number of DOFs for $\boldsymbol{p}$ and $\boldsymbol{\lambda}$ is the number of elements of meshes, and the number of DOFs for u is the number of nodes of meshes. Based on the results shown in Table 6.2, we find that the convergence rates for $(\boldsymbol{p}_h, u_h, \boldsymbol{\lambda}_h)$ are given by

$$\begin{aligned} &||\boldsymbol{p}-\boldsymbol{p}_h||_0 \sim \mathcal{O}(h), \quad ||\boldsymbol{\lambda}-\boldsymbol{\lambda}_h||_0 \sim \mathcal{O}(h), \\ &||u-u_h||_1 \sim \mathcal{O}(h), \quad ||u-u_h||_0 \sim \mathcal{O}\left(h^2\right). \end{aligned}$$

Remember that we are using the piecewise constant finite elements for discretizing $\boldsymbol{p}$ and $\boldsymbol{\lambda}$ and the first-order Lagrange finite elements for discretizing u. This means that expected optimal convergence rates are obtained for all variables.

Table 6.3 shows iteration counts for the Newton method with the block preconditioners $\mathcal{B}_h(\boldsymbol{r})$ and $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$ for different mesh sizes. We see from the relatively consistent iteration counts that both the exact and the inexact preconditioners are robust with respect to the mesh size. This demonstrates the optimality of the linear solver and the efficiency of the preconditioners. Compared with using the exact block preconditioners, using the inexact one results in a slight degradation in performance but nothing significant and negligible.

TABLE 6.1
Mesh sizes and numbers of DOFs.

Mesh	h	DOFs for $\boldsymbol{p}_h$	DOFs for u_h	DOFs for $\boldsymbol{\lambda}_h$	Total DOFs
$\mathcal{T}_1$	6.25e-02	1,024	289	1,024	2,337
$\mathcal{T}_2$	3.13e-02	4,096	1,089	4,096	9,281
$\mathcal{T}_3$	1.56e-02	16,384	4,225	16,384	36,993
$\mathcal{T}_4$	7.81e-03	65,536	16,641	65,536	147,713

TABLE 6.2
Errors and convergence rates for $(\boldsymbol{p}_h, u_h, \boldsymbol{\lambda}_h)$ (Example 6.1).

h	$\|\|\boldsymbol{p}-\boldsymbol{p}_h\|\|$	Order	$\|\|\boldsymbol{\lambda}-\boldsymbol{\lambda}_h\|\|$	Order
6.25e-02	2.17585e-01	—	8.95410e-02	—
3.13e-02	1.08967e-01	1.00	4.52978e-02	0.98
1.56e-02	5.45105e-02	1.00	2.27351e-02	1.00
7.81e-03	2.72596e-02	1.00	1.13809e-02	1.00

h	$\|\|u-u_h\|\|_1$	Order	$\|\|u-u_h\|\|$	Order
6.25e-02	2.17595e-01	—	7.97886e-03	—
3.13e-02	1.08968e-01	1.00	2.02665e-03	1.98
1.56e-02	5.45107e-02	1.00	5.12786e-04	1.98
7.81e-03	2.72596e-02	1.00	1.32618e-04	1.95

TABLE 6.3
Iteration counts for the Newton method with the block preconditioners $\mathcal{B}_h(\boldsymbol{r})$ and $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$ (Example 6.1).

mesh	$\mathcal{B}_h(\boldsymbol{r})$	$\widetilde{\mathcal{B}}_h(\boldsymbol{r})$
	$N_{\text{Newton}}(N_{\text{MINRES}})$	$N_{\text{Newton}}(N_{\text{MINRES}})$
$\mathcal{T}_1$	5(21)	5(26)
$\mathcal{T}_2$	5(20)	5(25)
$\mathcal{T}_3$	5(20)	5(25)
$\mathcal{T}_4$	5(19)	5(25)

TABLE 6.4
Iteration counts for the Picard method with the block preconditioners $\mathcal{B}_h(\boldsymbol{r})$ and $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$ (Example 6.1).

mesh	$\mathcal{B}_h(\boldsymbol{r})$	$\widetilde{\mathcal{B}}_h(\boldsymbol{r})$
	$N_{\text{Picard}}(N_{\text{MINRES}})$	$N_{\text{Picard}}(N_{\text{MINRES}})$
$\mathcal{T}_1$	36(19)	36(24)
$\mathcal{T}_2$	33(19)	33(24)
$\mathcal{T}_3$	30(18)	30(24)
$\mathcal{T}_4$	26(18)	26(23)

For comparison, we present the corresponding results for the Picard method in Table 6.4. Observe that comparing it with Table 6.3, the linear iteration numbers of the MINRES method for these two methods are almost the same, but the nonlinear iteration numbers of the Newton method are much less than those of the Picard method. Using mesh $\mathcal{T}_4$, we display the convergence histories of the Picard method and the Newton method in Figure 6.1. We see that the Newton method shows a dramatically faster convergence than the Picard method.

In Table 6.5, we further give the iteration numbers of MINRES at each Newton step for the exact and inexact preconditioners. From the results, we see that the iteration numbers of MINRES are almost invariant with different mesh sizes and iteration numbers. This verifies that our preconditioners are robust with respect to the iterative steps and mesh sizes. Besides, we again observe that the use of the inexact preconditioners has nearly no impact on the needed number of iterations for MINRES. Using the grid $\mathcal{T}_4$, we plot the convergence histories of the MINRES method at each Newton step in Figure 6.2. It can be seen that the relative residual decrease rapidly as we expected, which indicates that our preconditioners are effective.

Finally, we investigate the robustness of the block preconditioners with respect to the parameters α and β. We fix the mesh $\mathcal{T}_3$ and vary the parameters. The results

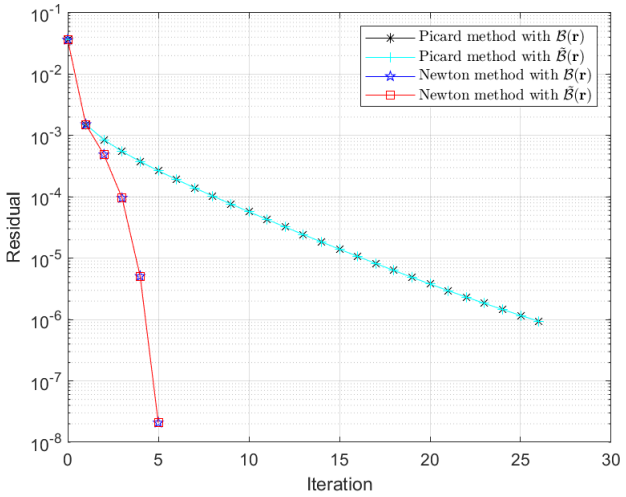

FIG. 6.1. *Convergence histories of the Picard method and the Newton method (Example* 6.1*).*

TABLE 6.5
Number of MINRES iterations with the block preconditioners $\mathcal{B}_h(\boldsymbol{r})$ *and* $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$ *at each Newton step (Example* 6.1*).*

mesh	$\mathcal{B}_h(\boldsymbol{r})$						$\widetilde{\mathcal{B}}_h(\boldsymbol{r})$				
	1	2	3	4	5		1	2	3	4	5
$\mathcal{T}_1$	15	18	22	24	25		17	23	26	31	33
$\mathcal{T}_2$	13	18	22	24	25		17	22	26	31	31
$\mathcal{T}_3$	13	17	20	23	25		18	22	26	30	31
$\mathcal{T}_4$	13	17	19	22	25		17	22	26	29	31

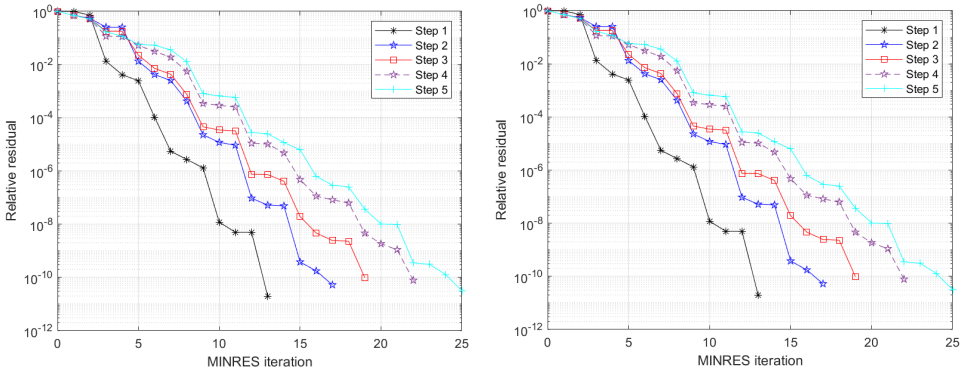

FIG. 6.2. *Convergence histories of the preconditioned MINRES method at each Newton step with* $\mathcal{B}_h(\boldsymbol{r})$ *(left) and* $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$ *(right).*

for the Newton method with the exact and inexact preconditioners are shown in Tables 6.6–6.7. We can see that the proposed preconditioners are very robust with respect to the parameters and that the use of the inexact preconditioner has nearly no impact on the required iterations for MINRES.

Example 6.2. In this example, we consider a nonsmooth problem. Let $\Omega = (0,1)^2$, and denote its center by $x_\Omega := (0.5, 0.5)$. Write $B_r(x_\Omega) = \left\{x \in \mathbb{R}^2 : ||x - x_\Omega||_{l_2} < r\right\}$

TABLE 6.6
Iteration counts for the Newton method with the block preconditioner $\mathcal{B}_h(\boldsymbol{r})$ (Example 6.1).

	α					
β	$N_{\text{Newton}}(N_{\text{MINRES}})$	1e5	1e3	1	1e-3	1e-5
	1	7(8)	7(10)	5(20)	2(41)	1(21)
	1e-3	14(10)	13(13)	8(31)	2(28)	2(28)
	1e-5	13(9)	13(10)	10(30)	2(22)	3(29)

TABLE 6.7
Iteration counts for the Newton method with the block preconditioner $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$ (Example 6.1).

	α					
β	$N_{\text{Newton}}(N_{\text{MINRES}})$	1e5	1e3	1	1e-3	1e-5
	1	7(12)	7(13)	5(25)	2(41)	1(21)
	1e-3	14(36)	13(33)	8(34)	2(29)	2(28)
	1e-5	13(23)	13(24)	10(34)	2(24)	3(30)

TABLE 6.8
Errors and convergence rates for u_h (Example 6.2).

h	$\|\|u-u_h\|\|$	Order	$\|\|u-I_hu\|\|$	Order
6.25e-02	1.12395e-01	—	1.17827e-01	—
3.13e-02	7.94646e-02	0.50	8.35176e-02	0.50
1.56e-02	6.10573e-02	0.38	6.33701e-02	0.40
7.81e-03	4.48697e-02	0.44	4.17839e-02	0.60

TABLE 6.9
Iteration counts for the Newton method with the block preconditioners $\mathcal{B}_h(\boldsymbol{r})$ and $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$ (Example 6.2).

mesh	$\mathcal{B}_h(\boldsymbol{r})$	$\widetilde{\mathcal{B}}_h(\boldsymbol{r})$
	$N_{\text{Newton}}(N_{\text{MINRES}})$	$N_{\text{Newton}}(N_{\text{MINRES}})$
$\mathcal{T}_1$	21(35)	21(35)
$\mathcal{T}_2$	23(32)	23(32)
$\mathcal{T}_3$	40(31)	34(31)
$\mathcal{T}_4$	21(29)	24(31)

with $r = 1/3$; the function f is chosen as a characteristic function $f \coloneqq \chi_{B_r(x_\Omega)}$. Let I_h be the standard nodal interpolation operator to U_h; the initial guess u_h^0 is taken as $u_h^0 \coloneqq I_h f$, and $\left(p_h^0, \boldsymbol{\lambda}_h^0\right)$ is computed by the equations (3.1).

First, we aim to investigate the convergence rates. From [39], if $\alpha = 0.02$, the exact solution to the model (1.2) is given by

$$u = \begin{cases} 1 - \frac{2\alpha}{r} = 0.94 & x \in B_r(x_\Omega), \\ \frac{2\pi r\alpha}{1-\pi r^2} \approx 0.03 & x \in \Omega \backslash B_r(x_\Omega). \end{cases}$$

We perform numerical tests for the primal-dual finite element discretization to (2.2) with $\beta = 1\text{e}{-5}$. The errors and convergence rates for u are displayed in Table 6.8. Note that the exact solution $u \in L^\infty(\Omega)$; thus, the convergence rates are not as perfect as the ones in Example 6.1. Even so, the numerical results are in accord with the theoretical results in Proposition 10.9 of [4] for the standard finite element discretization to (1.4). The error estimate of primal-dual finite element discretization is left for further work; we also refer the reader to [14, 15, 16, 28, 45] for some discussions on this direction.

In Table 6.9, we present the iteration numbers of the Newton method on various meshes. As predicted from the analysis, the numbers of MINRES iterations are stable

when we vary the mesh size h. Besides, the iteration numbers of the MINRES method with $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$ are almost the same as those with $\mathcal{B}_h(\boldsymbol{r})$. This is expected, and the difference is by no means significant. One can see the phenomenon that the number of Newton iterations with $\mathcal{B}_h(\boldsymbol{r})$ is less than that with $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$ in $\mathcal{T}_3$, but it is the other way around in $\mathcal{T}_4$. The reason we think so is that some Newton directions computed with $\mathcal{B}_h(\boldsymbol{r})$ may not be as good as the ones by $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$ in $\mathcal{T}_3$ for such a nonsmooth problem. Overall, we can conclude that our preconditioners are effective and robust with respect to the mesh size h. Figure 6.3 plots the iteration numbers of MINRES at each Newton step. Again, we observe that the iteration numbers of MINRES are around a constant value for different Newton iterations. This confirms the robustness of the proposed preconditioners with respect to the iterative step. Specifically, Figure 6.4 shows the damping parameter with respect to the iterative step on mesh $\mathcal{T}_4$. We find that the

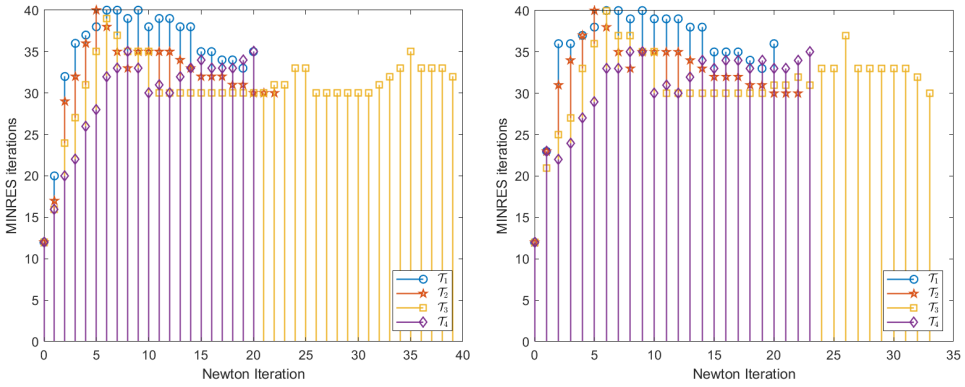

FIG. 6.3. *Number of MINRES iterations with the block preconditioners* $\mathcal{B}_h(\boldsymbol{r})$ *(left) and* $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$ *(right) at each Newton step (Example* 6.2*).*

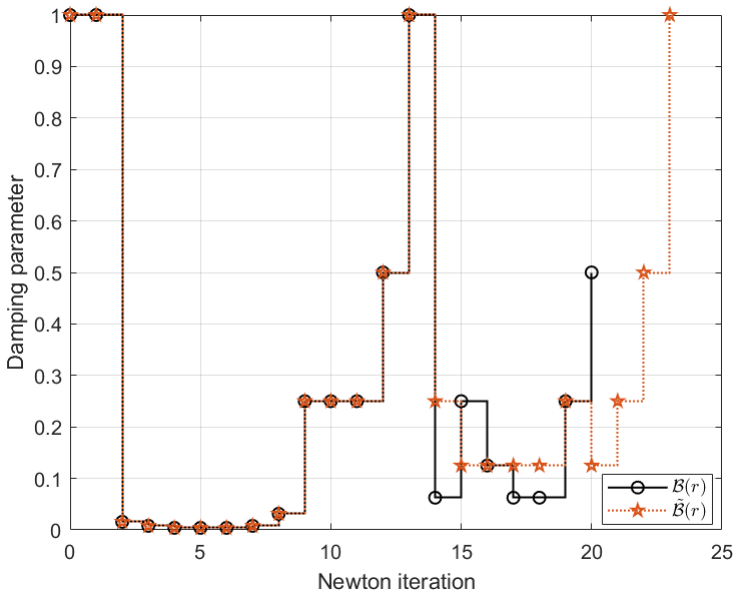

FIG. 6.4. *Damping parameter at each Newton iteration with* $\mathcal{B}_h(\boldsymbol{r})$ *and* $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$ *on mesh* $\mathcal{T}_4$.

damping parameter θ is very small at some iterative steps. The damping strategy is crucial to guarantee the global convergence of the Newton method for this example. It may diverge if we do not use this damping strategy.

As in the previous example, we vary the parameters α and β to study the robustness of the preconditioners. Tables 6.10–6.11 show the results for the Newton method with the exact and inexact preconditioners on mesh $\mathcal{T}_3$. Again, we see that the preconditioners show relative robustness with respect to the parameters. The inexact preconditioner requires a slightly higher number of iterations to converge compared to the exact one, as we saw in the previous example.

For comparison, we present the number of iterations required by Picard method for different α and β on mesh $\mathcal{T}_3$ in Tables 6.12–6.13. Compared with the results of Newton method, we can see that the Newton iteration converges rapidly than the Picard iteration for the considered parameters. For the case of $\alpha = 1\mathrm{e}-2, \beta = 1\mathrm{e}-3$, Figure 6.5 plots the convergence histories of Picard method and Newton method with $\mathcal{B}_h(\boldsymbol{r})$ and $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$. As we conclude, the Newton iteration behaves similarly to the Picard iteration in the early stages but converges rapidly in the end stages.

TABLE 6.10
Iteration counts for the Newton method with the block preconditioner $\mathcal{B}_h(\boldsymbol{r})$ (Example 6.2).

	α				
	$N_{\text{Newton}}(N_{\text{MINRES}})$	1e-1	5e-2	1e-2	5e-3
β	1	22(28)	8(35)	5(46)	4(49)
	1e-1	31(26)	6(31)	10(43)	9(45)
	1e-2	27(21)	7(28)	9(40)	12(41)
	1e-3	10(18)	8(25)	15(39)	18(37)

TABLE 6.11
Iteration counts for the Newton method with the block preconditioner $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$ (Example 6.2).

	α				
	$N_{\text{Newton}}(N_{\text{MINRES}})$	1e-1	5e-2	1e-2	5e-3
β	1	22(33)	8(42)	5(46)	4(49)
	1e-1	31(30)	6(35)	10(44)	9(45)
	1e-2	37(25)	7(30)	9(42)	12(41)
	1e-3	10(22)	8(27)	18(39)	18(37)

TABLE 6.12
Iteration counts for the Picard method with the block preconditioner $\mathcal{B}_h(\boldsymbol{r})$ (Example 6.2).

	α				
	$N_{\text{Picard}}(N_{\text{MINRES}})$	1e-1	5e-2	1e-2	5e-3
β	1	20(27)	26(31)	13(44)	9(50)
	1e-1	39(22)	37(27)	18(37)	12(42)
	1e-2	68(19)	52(24)	23(31)	16(35)
	1e-3	90(18)	71(22)	29(29)	20(32)

TABLE 6.13
Iteration counts for the Picard method with the block preconditioner $\widetilde{\mathcal{B}}_h(\boldsymbol{r})$ (Example 6.2).

	α				
	$N_{\text{Picard}}(N_{\text{MINRES}})$	1e-1	5e-2	1e-2	5e-3
β	1	20(33)	26(39)	13(45)	9(50)
	1e-1	39(28)	37(34)	18(42)	12(45)
	1e-2	68(24)	52(31)	23(38)	16(42)
	1e-3	90(22)	71(29)	29(36)	20(39)

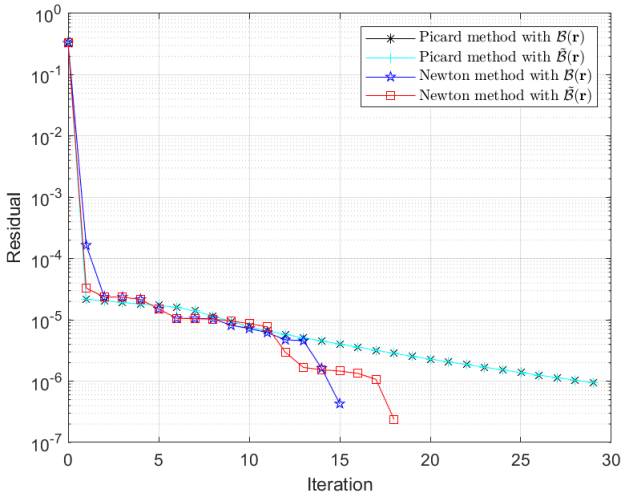

FIG. 6.5. *Convergence histories of the Picard method and the Newton method (Example* 6.2*).*

TABLE 6.14
Iteration counts for the Picard method and the Newton method (Example 6.3*).*

p	$N_{\text{Picard}}(N_{\text{MINRES}})$	$N_{\text{Newton}}(N_{\text{MINRES}})$
1	66(29)	16(41)
2	67(30)	15(39)
∞	86(29)	18(43)

Example 6.3. In this example, we consider a benchmark problem as in [3, 5, 41]. Let $\Omega = (0,1)^2$, and denote its center by x_Ω. Given a triangulation $\mathcal{T}_h$ of Ω, we take the noise function $\xi_h \in U_h$, whose coefficient vector is some random numbers from a normal distribution with zero mean and standard deviation 1. Let $B_r^p(x_\Omega) = \{x \in \mathbb{R}^d : |x - x_\Omega|_{l_p} < r\}$ with $r = 1/3$, $p \in [1, \infty]$, and take f_0 as the characteristic function $f_0 := \chi_{B_r^p(x_\Omega)}$. We then define

$$f = f_0 + \delta\xi_h \quad \text{with} \quad \delta = 0.1.$$

The parameters' values are $\alpha =$ 5e-2 and $\beta =$ 1e-3. The initial value for the Picard method is set as in Example 6.2, while for the Newton method, the initial value is produced by five iterations of the Picard method. Namely, when using the Newton method, we first carry out five iterations of the Picard method and then switch to using the Newton method.

To experimentally study the effectiveness of the proposed method, we run the Picard method and the Newton method with $p = 1, 2, \infty$ on mesh $\mathcal{T}_4$. Figure 6.6 displays the input noise image data and the outputs of the iterative schemes. Notice that the Newton method and the Picard method yield very similar results. However, the Newton method converges much faster; see Table 6.14 and Figure 6.7. From the outputs, we see that noise is removed effectively. The boundary is slightly smoothed, especially the corners. The numerical results indeed show the inherited properties of the ROF model.

The iteration numbers displayed in Table 6.14 for the Picard method and the Newton method are produced with preconditioner $\widetilde{\mathcal{B}_h}(\boldsymbol{r})$. The convergence histories are further demonstrated in Figure 6.7. From the result, we see that the Newton method converges much faster than the Picard method after a few damped Newton updatings.

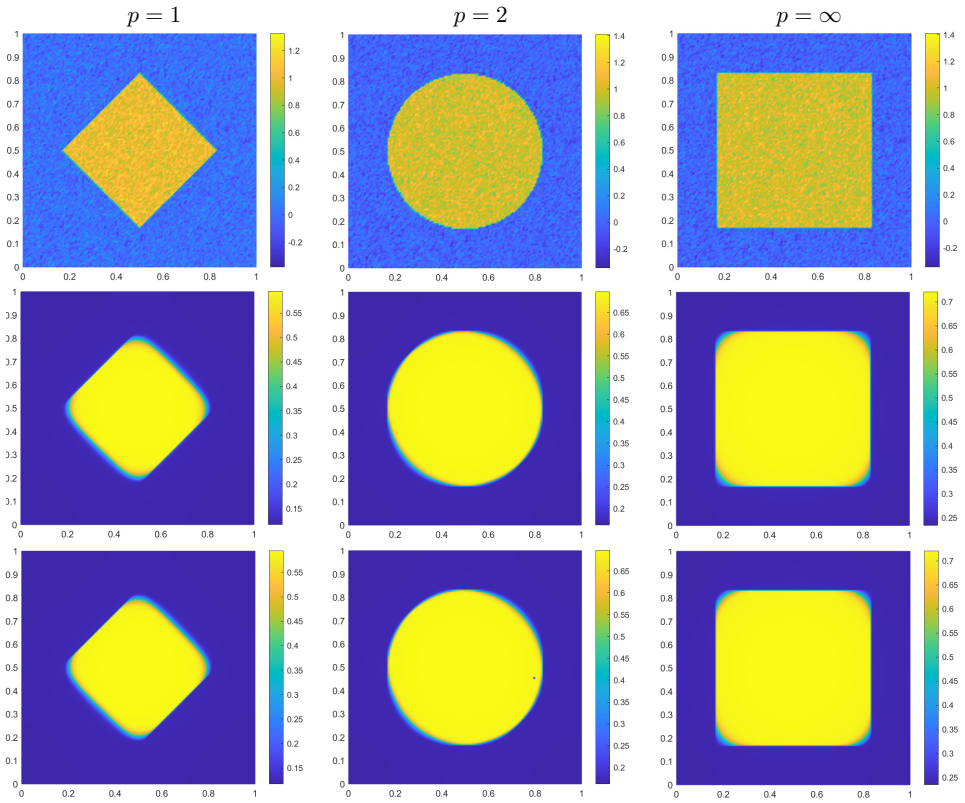

FIG. 6.6. *Noisy image, denoised image with the Picard method, and denoised image with the Newton method (from top to bottom).*

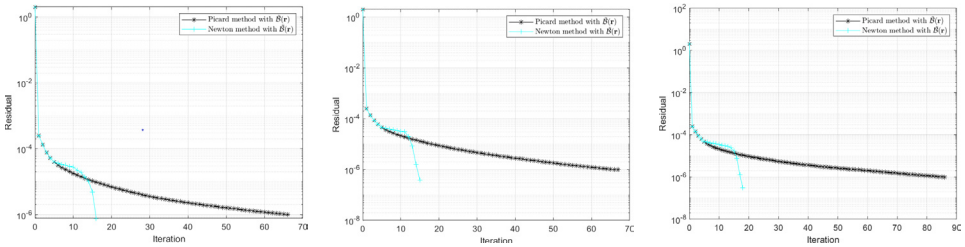

FIG. 6.7. *Convergence histories of the Picard method and the Newton method for* $p = 1, 2, \infty$ *(from left to right) (Example* 6.3*).*

7. Conclusions. In this paper, we propose a preconditioned Newton solver for primal-dual finite element approximation of total variation minimization and minimum-surface problems. We develop some block diagonal preconditioners for the discrete problems at each Newton iteration, which are robust with respect to the mesh size, the penalization parameter, the regularization parameter, and the iterative step. We further prove that the resulting preconditioned MINRES converges uniformly. The theoretical findings are demonstrated by numerical experiments. In this work, we only use the proposed method to compute a benchmark example on

image denoising. It is interesting to extend it to image inpainting and deblurring problems, and we leave these as subjects of future endeavor.

Acknowledgment. We gratefully acknowledge the anonymous referees for their pertinent and perceptive comments which have significantly improved our exposition.

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] I. Babuška and A. K. Aziz, *Survey lectures on the mathematical foundations of the finite element method*, in The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations, Academic Press, New York, 1972, pp. 1–359.

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

[4] S. Bartels, *Numerical Methods for Nonlinear Partial Differential Equations*, Springer Series in Computational Mathematics 47, Springer-Verlag, Berlin, 2015.

[5] S. Bartels, L. Diening, and R. H. Nochetto, *Unconditional stability of semi-implicit discretizations of singular flows*, SIAM J. Numer. Anal., 56 (2018), pp. 1896–1914.

[6] S. Bartels, R. H. Nochetto, and A. J. Salgado, *Discrete total variation flows without regularization*, SIAM J. Numer. Anal., 52 (2014), pp. 363–385.

[7] S. Bartels, R. H. Nochetto, and A. J. Salgado, *A total variation diminishing interpolation operator and applications*, Math. Comp., 84 (2015), pp. 2569–2587.

[8] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, 3rd ed., Texts in Applied Mathematics 15, Springer-Verlag, Berlin, 2008.

[9] F. Brezzi, *On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers*, Rev. Française Automat. Informat. Recherche Opérationnelle Sér. Rouge, 8 (1974), pp. 129–151.

[10] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer Series in Computational Mathematics 15, Springer-Verlag, Berlin, 1991.

[11] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, 2000.

[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] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vision, 20 (2004), pp. 89–97.

[14] A. Chambolle, S. E. Levine, and B. J. Lucier, *An upwind finite-difference method for total variation-based image smoothing*, SIAM J. Imaging Sci., 4 (2011), pp. 277–299.

[15] A. Chambolle and T. Pock, *Crouzeix-Raviart approximation of the total variation on simplicial meshes*, J. Math. Imaging Vision, 62 (2020), pp. 872–899.

[16] A. Chambolle and T. Pock, *Approximating the total variation with finite differences or finite elements*, in Geometric Partial Differential Equations, Part II, Handbook of Numerical Analysis 22, Elsevier–North-Holland, Amsterdam, 2021, pp. 383–417.

[17] 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.

[18] T. F. Chan and J. Shen, *On the role of the BV image model in image restoration*, in Recent Advances in Scientific Computing and Partial Differential Equations (Hong Kong, 2002), Contemporary Mathematics 330, American Mathematical Society, Providence, RI, 2003, pp. 25–41.

[19] L. Chen, *iFEM: An Integrated Finite Element Methods Package in MATLAB*, Technical report, University of California at Irvine, 2009.

[20] D. C. Dobson and C. R. Vogel, *Convergence of an iterative method for total variation denoising*, SIAM J. Numer. Anal., 34 (1997), pp. 1779–1791.

[21] V. Dolean, P. Jolivet, and F. Nataf, *An Introduction to Domain Decomposition Methods*, Society for Industrial and Applied Mathematics, Philadelphia, 2015.

[22] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM J. Imaging Sci., 2 (2009), pp. 323–343.

[23] W. Hackbusch, *Iterative Solution of Large Sparse Systems of Equations*, 2nd ed., Applied Mathematical Sciences 95, Springer-Verlag, Berlin, 2016.

[24] M. Herrmann, R. Herzog, H. Kröner, S. Schmidt, and J. Vidal, *Analysis and an interior-point approach for TV image reconstruction problems on smooth surfaces*, SIAM J. Imaging Sci., 11 (2018), pp. 889–922.
[25] M. Herrmann, R. Herzog, S. Schmidt, J. Vidal-Núñez, and G. Wachsmuth, *Discrete total variation with finite elements and applications to imaging*, J. Math. Imaging Vision, 61 (2019), pp. 411–431.
[26] M. Hintermüller and K. Kunisch, *Total bounded variation regularization as a bilaterally constrained optimization problem*, SIAM J. Appl. Math., 64 (2004), pp. 1311–1333.
[27] Q. Hu, X. Tai, and R. Winther, *A saddle point approach to the computation of harmonic maps*, SIAM J. Numer. Anal., 47 (2009), pp. 1500–1523.
[28] M.-J. Lai and L. Matamba Messi, *Piecewise linear approximation of the continuous Rudin-Osher-Fatemi model for image denoising*, SIAM J. Numer. Anal., 50 (2012), pp. 2446–2466.
[29] D. Lao and S. Zhao, *Fundamental Theories and Their Applications of the Calculus of Variations,* Beijing Institute of Technology Press, Beijing, 2021.
[30] C.-O. Lee, E.-H. Park, and J. Park, *A finite element approach for the dual Rudin-Osher-Fatemi model and its nonoverlapping domain decomposition methods*, SIAM J. Sci. Comput., 41 (2019), pp. B205–B228.
[31] D. Liberzon, *Calculus of Variations and Optimal Control Theory*, Princeton University Press, Princeton, NJ, 2012.
[32] J. Lottes, *Towards Robust Algebraic Multigrid Methods for Nonsymmetric Problems*, Doctoral thesis, Oxford University, 2017.
[33] K.-A. Mardal and R. Winther, *Preconditioning discretizations of systems of partial differential equations*, Numer. Linear Algebra Appl., 18 (2011), pp. 1–40.
[34] A. Marquina and S. Osher, *Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal*, SIAM J. Sci. Comput., 22 (2000), pp. 387–405.
[35] J. D. Moore, *Introduction to global analysis: Minimal surfaces in Riemannian manifolds*, Bull. Amer. Math. Soc., 57 (2020), pp. 353–356.
[36] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Applied Mathematical Sciences 153, Springer-Verlag, Berlin, 2003.
[37] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
[38] V. V. Shaĭdurov, *Multigrid Methods for Finite Elements*, Mathematics and Its Applications 318, Kluwer Academic Publishers, Norwell, MA, 1995.
[39] D. Strong and T. Chan, *Edge-preserving and scale-dependent properties of total variation regularization*, Inverse Problems, 19 (2003), pp. S165–S187.
[40] X.-C. Tai and C. Wu, *Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model*, in Scale Space and Variational Methods in Computer Vision, X.-C. Tai, K. Mørken, M. Lysaker, and K.-A. Lie, eds., Springer-Verlag, Berlin, 2009, pp. 502–513.
[41] W. Tian and X. Yuan, *Convergence analysis of primal-dual based methods for total variation minimization with finite element approximation*, J. Sci. Comput., 76 (2018), pp. 243–274.
[42] A. Toselli and O. Widlund, *Domain Decomposition Methods—Algorithms and Theory*, Springer Series in Computational Mathematics 34, Springer-Verlag, Berlin, 2005.
[43] P. S. Vassilevski, *Multilevel Block Factorization Preconditioners*, Springer-Verlag, Berlin, 2008.
[44] C. R. Vogel and M. E. Oman, *Iterative methods for total variation denoising*, SIAM J. Sci. Comput., 17 (1996), pp. 227–238.
[45] 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.
[46] C. Wu and X.-C. Tai, *Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models*, SIAM J. Imaging Sci., 3 (2010), pp. 300–339.
[47] J. Xu, X.-C. Tai, and L.-L. Wang, *A two-level domain decomposition method for image restoration*, Inverse Probl. Imaging, 4 (2010), pp. 523–545.
[48] J. Xu and L. Zikatanov, *Algebraic multigrid methods*, Acta Numer., 26 (2017), pp. 591–721.
[49] C. H. Yao, *Finite element approximation for TV regularization*, Int. J. Numer. Anal. Model., 5 (2008), pp. 516–526.