<table>
<thead>
<tr>
<th>Title</th>
<th>Identification of discontinuous coefficients in elliptic problems using total variation regularization (Published version)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Chan, Tony F.; Tai, Xue Cheng</td>
</tr>
<tr>
<td>Date</td>
<td>2003</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10220/4605">http://hdl.handle.net/10220/4605</a></td>
</tr>
</tbody>
</table>
IDENTIFICATION OF DISCONTINUOUS COEFFICIENTS IN ELLIPTIC PROBLEMS USING TOTAL VARIATION REGULARIZATION∗

TONY F. CHAN† AND XUE-CHENG TAI‡

Abstract. We propose several formulations for recovering discontinuous coefficients in elliptic problems by using total variation (TV) regularization. The motivation for using TV is its well-established ability to recover sharp discontinuities. We employ an augmented Lagrangian variational formulation for solving the output-least-squares inverse problem. In addition to the basic output-least-squares formulation, we introduce two new techniques for handling large observation errors. First, we use a filtering step to remove as much of the observation error as possible. Second, we introduce two extensions of the output-least-squares model; one model employs observations of the gradient of the state variable while the other uses the flux. Numerical experiments indicate that the combination of these two techniques enables us to successfully recover discontinuous coefficients even under large observation errors.

Key words. elliptic, inverse problems, parameter estimation, total variational norm, regularization, noise removal

AMS subject classifications. 65K, 65M30, 49B

DOI. 10.1137/S1064827599326020

1. Introduction. Consider the partial differential equation

\begin{equation}
\begin{cases}
-\nabla \cdot (q(x)\nabla u) = f & \text{in } \Omega \subset \mathbb{R}^2, \\
u = 0 & \text{on } \partial \Omega.
\end{cases}
\end{equation}

We want to use observations of the solution \(u\) to recover the coefficient \(q(x)\). We shall especially treat the case that \(q(x)\) has discontinuities. This problem arises in many industrial applications, for example, reservoir simulations [24], underground water investigations [50, 19, 47], and other applications [10, 22, 43].

In this article, we shall consider the following three kinds of observation data for identifying the coefficient:

1. We have an observation \(u_d \in L^2(\Omega)\) for the solution \(u\). We note that this condition is weaker than required of the solution \(u\).
2. We have observations \(u_d \in L^2(\Omega), \overline{u}_g \in (L^2(\Omega))^2\) for the solution \(u\) and its gradient, respectively.
3. We have observations \(u_d \in L^2(\Omega), \overline{u}_v \in (L^2(\Omega))^2\) for the solution \(u\) and the velocity \(q\nabla u\), respectively.

Equation (1.1) can be used to describe the one-phase groundwater motion with \(u\) representing the piezometric head of groundwater in \(\Omega\); the function \(f\) characterizes the sources and sinks in \(\Omega\). The filtration (transmissivity) coefficient \(q\) is, physically,
positive and piecewise smooth with possible discontinuities on some interfaces in $\Omega$. Equation (1.1) can also be used to describe the fluid flow of a one-phase reservoir. In such a case, $u$ is the pressure related to the flow in a heterogeneous reservoir; $q$ is called the absolute permeability, which is related to the permeability of the medium and other physical parameters such as the viscosity of the fluid. For a one-phase reservoir, the first estimation problem corresponds to measuring the pressure to recover the absolute permeability $q$. The second estimation problem corresponds to measuring both the pressure and its gradient to recover $q$. For the third kind of observation, we need to measure both the pressure and the velocity of the fluid to recover $q$. One application of the proposed algorithms is the upscaling problem for oil reservoirs [39, 26], where the observation data is essentially available at every point. For other practical applications, the observation is often measured at certain points and we need to interpolate the point observations to get distributed observations. However, it is well known that the mesh size approximating the coefficient $q(x)$ depends on the amount of information we could measure; see Yeh [50], Kunisch and White [36], Tai and Karkainen [45], and Tai and Neittaanmaki [46]. If we only have measurements at a limited number of points, then we need to use a very coarse mesh in the parameterization. Correspondingly, we may not be able to recover the fine structures of the coefficients. In this work, we assume that we have an observation on a mesh as fine as needed to resolve $q$ and we want to recover both the values and the locations of the discontinuities of the coefficients. Numerical evidence shows that we need rather fine observations if we want to recover the locations of the discontinuities. In practical applications, the gradients may not be measurable directly and we may be forced to use the differentiation formulas as in Anderssen and Hegland [3] to get the gradients. For smooth functions, the gradients obtained as in [3] do not explode as the mesh size goes to zero even in the case where there is observation error. For interface problems, the gradients are discontinuous and it will be interesting to test how much noise we can handle for the inverse problem by using the differentiation formula of [3]. In this work, we only assume the coefficient $q$ is piecewise smooth but otherwise completely general. In the case that $q$ can be well approximated by a piecewise smooth function with a small number of piecewise smooth regions, one could also use the level set method to represent the geometry of the discontinuities directly, as in Chan and Tai [14].

In the case where the gradient can be measured exactly, the inverse problem is reduced to solve $q$ from the following hyperbolic conservation law:

\[
\text{div}(\vec{\beta} q) = f, \quad \vec{\beta} = \nabla u. \tag{1.2}
\]

Such a problem has many important applications, for example, in the semiconductor industry, and its numerical solution is not a trivial matter [4, 28]. Using the special property that the vector $\vec{\beta}$ is the gradient, Bank et al. proposed special upwinding techniques [4] to get an accurate solution $q$ without artificial oscillations. The algorithms proposed here offer another stabilized least-square-type of method for solving (1.2) that can also handle the case where the vector $\vec{\beta}$ may contain large errors.

It is well known that inverse problems are ill-posed. The common approach is to use the output-least-squares method with certain regularization techniques [33, 1, 6]. When the coefficient $q(x)$ is assumed to be smooth, successful numerical methods are available; see Ito and Kunisch [27] and Kunisch and Tai [35]. However, when the coefficients have discontinuities, the problem is much more difficult. In addition, we also want to handle large observation noise. Detecting discontinuity of the coefficient
has long been an interesting research topic; see [9, 10, 8, 32, 20, 21, 18, 22, 30, 31, 42] for some earlier results for different kind of equations. The work of Chow and Anderssen [17] is closely related to ours. In [17], a linear functional strategy is extended to determine the zonation structure of $q(x)$. In this work, we are interested in using interior measurement to detect discontinuity and also recover the coefficient value. In Ben Ameur, Chavent, and Jaffr´e [5], an adaptive refinement strategy is used to recover a piecewise constant permeability coefficient. In the case where boundary measurements are available for the solution and the flux, good numerical algorithms have been developed; see [7, 22, 16] and references therein.

The main concern of this work is developing numerical methods that can handle discontinuous coefficients and can treat observations containing large random errors. The contribution of this work can be classified into three points. First, we introduce a presmoothing method that removes random noise from the observations. Numerical tests and also a mathematical observation reveal that this presmoothing is very important in the case where the observation errors are large and a small mesh size is used for the approximation of the coefficient $q$ and the state $u$. Second, we introduce two new output-least-squares models. Inverse problems have been traditionally addressed using observations of the state $u$. In this work, we propose to observe also the gradient of $u$ or the flux $q\nabla u$. Numerical experiments indicate that the gradient observations can tolerate much larger observation errors. The practical implication is that we shall measure the gradient whenever this is possible even though the measurement for the gradient may contain large errors. The third contribution lies in the use of TV-norm regularization for the coefficient. The use of the TV-norm regularization is motivated by its success in image processing [2, 12, 11, 41, 40, 49]. Compared with the intensive studies of the TV-norm for image processing, the use of TV-norm for parameter estimation problems for partial differential equations is rather sparse. There are very few works; see [48, 22, 33, 15, 29]. Here, we further develop this technique in combination with the presmoothing method which enables us to identify discontinuous coefficients with very large observation errors. In section 6.1.4, we give an example which will show that the TV-norm regularization works better than the traditional $H^1$- or $H^2$- norm regularization; see Figure 8. See also Lin and Ramirez [37] for some numerical tests which show that the usual regularization method smears out the sharp jumps, or introduces Gibbs oscillations, at the locations of the discontinuities.

In solving the output-least-squares minimization problem, we employ an augmented Lagrangian variational formulation by which the inverse problem is transformed into a nonlinear minimization problem in both $u$ and $q$. This approach has been quite successful in inverse parameter estimation problems with smooth coefficients; see for example [27, 35]. The algorithms use several parameters (i.e., the penalization parameter, the regularization parameter, and the TV-norm approximation parameter $\epsilon$) that need to be chosen properly. In the present work, we shall concentrate on the performance of the proposed algorithms instead of on the optimal automatic determination of these parameters [19, 38].

The organization of the paper is as follows. In the next section, we present the three variants of the output-least-squares formulation of the inverse problems. In section 3, we define the corresponding augmented Lagrangian formulations. The noise removing techniques are introduced in section 4. Numerical results will be presented in sections 5. Some concluding remarks are given in section 6.
2. Formulation of the identification problems.

2.1. Output-least-squares approach. As the observations may contain large random observation errors, and also due to the lack of proper boundary conditions for the coefficient \( q(x) \), it is not preferable to use direct methods for solving the inverse problems. Instead, we shall search for a coefficient \( q(x) \) which produces a solution \( u(x) \) that has the smallest distance to the observations, subject to certain regularity conditions on \( q(x) \).

Let the set \( K \) denote the set of admissible coefficients and the equality \( e(q, u) = 0 \) represent the equation constraint (1.1) in a suitable space; see section 2.3. Finally, we use \( u(q) \) to denote the solution to \( e(q, u) = 0 \) for a given \( q \).

Corresponding to each of the three kinds of observations, we shall solve

\[
\begin{align*}
(P1) \quad \min_{e(q, u) = 0, \; q \in K} & \frac{1}{2} \| u(q) - u_d \|^2_{L^2(\Omega)} + \beta R(q), \\
(P2) \quad \min_{e(q, u) = 0, \; q \in K} & \frac{1}{2} \| u(q) - u_d \|^2_{L^2(\Omega)} + \frac{1}{2} \gamma \| \nabla u(q) - \bar{u}_g \|^2_{L^2(\Omega)} + \beta R(q), \\
(P3) \quad \min_{e(q, u) = 0, \; q \in K} & \frac{1}{2} \| u(q) - u_d \|^2_{L^2(\Omega)} + \frac{1}{2} \gamma \| q \nabla u(q) - \bar{u}_v \|^2_{L^2(\Omega)} + \beta R(q),
\end{align*}
\]

where \( R(q) \) is a regularization functional used to control the regularity of \( q(x) \). In all our experiments, we use \( \gamma = 1 \) for simplicity.

2.2. TV regularization. If the coefficient is continuous, \( R(q) = \| q \|^2_{H^2(\Omega)} \) or \( \| q \|^2_{H^1(\Omega)} \) is commonly used as the regularization term; see [33]. In [35], existence, uniqueness, and convergence have been proved for such kinds of regularization. However, if the coefficient has large jumps, the use of \( H^2 \)- or \( H^1 \)-regularization is not appropriate due to the discontinuities of the coefficient. In this work, we shall take as a regularization functional the following:

\[
TV(q) = \int_{\Omega} |\nabla q| \, dx,
\]

where \( TV(q) \) is the total variation of \( q \); see Ziemer [51] and Giusti [25] for definitions. When \( q \) is not differentiable, \( |\nabla q| \) is understood as a measure; see page 111 of [51].

However, the TV-norm functional is not differentiable with respect to \( q \). For numerical purposes, we introduce

\[
R(q) = \int_{\Omega} \sqrt{|\nabla q|^2 + \varepsilon} \, dx.
\]

This functional is well defined for \( q \in H^1(\Omega) \). However, for convenience, in our numerical simulations we use piecewise constants to approximate \( q \) and, consequently, \( q \) is not in \( H^1(\Omega) \) and \( R(q) \) is not well defined. In section 6, we shall show that there exists a modified version of \( R(q) \) that does approximate \( TV(q) \) as \( \varepsilon \) goes to zero.

2.3. The equation constraint. In the following, the admissible coefficients set \( K \) is taken to be

\[
K = \{ q \mid q \in L^\infty(\Omega) \cap TV(\Omega), \; 0 < k_1 \leq q(x) \leq k_2 < \infty \},
\]
with \( k_1 \) and \( k_2 \) known a priori. For any given \( q \in K \) and \( u \in H^2_0(\Omega) \), we shall define the linear operators \( A_q : H^1_0(\Omega) \to H^{-1}(\Omega) \), \( B_u : K \to H^{-1}(\Omega) \) by

\[
A_q u = -\nabla \cdot (q \nabla u),
\]

(2.2)

\[
B_u q = -\nabla \cdot (q \nabla u).
\]

(2.3)

For any \( q \in K \), \( A_q \) is a homomorphism from \( H^1_0(\Omega) \) to \( H^{-1}(\Omega) \). Let \( C \) be any homomorphism from \( H^{-1}(\Omega) \) to \( H^2_0(\Omega) \); e.g., in our algorithms we use the inverse Laplacian operator with homogeneous Dirichlet boundary conditions for \( C \). Moreover, we assume that \( C \) is chosen such that \( (C v, w) \), which is the dual action of \( w \) on \( Cv \), defines an inner product for \( v, w \in H^{-1}(\Omega) \). Correspondingly, the operator \( C \) should satisfy

\[
(C v, w) = (v, C w) \quad \forall v, w \in H^{-1}(\Omega),
\]

(2.4)

\[
(v, w)_{H^{-1}(\Omega)} = (C v, w) \quad \forall v, w \in H^{-1}(\Omega),
\]

(2.5)

\[
\|v\|^2_{H^{-1}(\Omega)} = (C v, v) \quad \forall v \in H^{-1}(\Omega).
\]

We now define the equation constraint as

\[
e(q, u) = C(-\nabla \cdot (q \nabla u) - f)
\]

= \( C(A_q u - f) \)

= \( C(B_u q - f) \).

(2.6)

In the discrete approximations, \( A_q, B_u \), and \( C \) are matrices and we shall use \( A^*_q \), \( B^*_u \), and \( C^* \) to denote the corresponding transposes. Let \( P_h \) and \( S_h \) be the finite element spaces for approximating \( q(x) \) and \( u(x) \), respectively (see [13]). Assume that \( \{\phi_i\}_{i=1}^{n_1} \) and \( \{\psi_i\}_{i=1}^{n_2} \) are the basis functions for \( P_h \) and \( S_h \), respectively. Then the matrix \( A_q \) corresponds to the following standard discrete elliptic operator:

\[
(A_q u, v) = (q \nabla u, \nabla v) \quad \forall u, v \in S_h \subset H^1_0(\Omega).
\]

For a given \( q \), the matrix \( A_q = [a_{ij}]_{n_1 \times n_1} \) has the entries \( a_{ij} = \int_\Omega q \nabla \psi_i \cdot \nabla \psi_j \). The matrix \( B_u \) now corresponds to

\[
(B_u q, v) = (\nabla u \cdot \nabla v, q) \quad \forall q \in P_h, \quad \forall v \in S_h.
\]

(2.7)

For a given \( u \), the matrix \( B_u = [b_{ij}]_{n_1 \times n_2} \) has the entries \( b_{ij} = \int_\Omega \phi_i \nabla u \cdot \nabla \psi_j \). It is easy to see that \( B_u \) is generally neither square nor symmetric. The matrix \( C \) is symmetric positive definite and corresponds to the homomorphism operator that maps \( H^{-1}(\Omega) \) to \( H^2_0(\Omega) \). The most straightforward homomorphism from \( H^{-1}(\Omega) \) to \( H^2_0(\Omega) \) is the inverse of the Laplace operator and this is what we use in our implementation. More specifically, let \( D \) be the discrete Laplacian matrix corresponding to

\[
(D u, v) = (\nabla u, \nabla v) \quad \forall u, v \in S_h \subset H^1_0(\Omega);
\]

then we take \( C = D^{-1} \). The primary aim in using this operator \( C \) is to measure the equation error in the \( H^{-1} \)-norm instead of \( L^2 \) or other norms, but it also has a preconditioning effect on the linear algebraic equations we need to solve later.
3. The augmented Lagrangian methods. For the equations we consider in this work, the equation constraint $e(q, u)$ is in a bilinear form. The augmented Lagrangian approach posed in [35] reduces the minimization problems to a system of coupled algebraic equations. Efficient iterative methods can be used to solve these equations. We shall treat each of the three variational formulations in turn.

3.1. Augmented Lagrangian for (P1). In this case, only $L^2$-observations are available. The augmented Lagrangian method is used to enforce the equation constraint $e(q, u) = 0$.

For any $r > 0$, let us define the augmented Lagrangian functional as

$$L_r(q, u, \lambda) = \frac{1}{2} \| u - u_d \|_{L^2(\Omega)}^2 + \frac{1}{2} \gamma \| \nabla u - \bar{u}_g \|_{L^2(\Omega)}^2 + \beta R(q) + \frac{r}{2} \| e(q, u) \|_{L^2(\Omega)}^2 + (\lambda, e(q, u))_{L^2(\Omega)}.$$ 

When the equation constraint takes the specific form given in section 2.3, the existence of a saddle point for $L_r$ is known; see [35]. Moreover, if $(\tilde{q}, \tilde{u}, \tilde{\lambda})$ is a saddle point for $L_r$, then $(\tilde{q}, \tilde{u})$ is a minimizer of $(P1)$. Later, we shall propose an iterative algorithm to find the saddle point.

3.2. Augmented Lagrangian for (P2). When we have $H^1$-observations, we shall enforce the equation constraint $-\nabla \cdot (q \nabla u) = f$ in the $H^{-1}$-norm that we introduced earlier in (2.4) and (2.5). The advantage of using the $H^{-1}$-norm is that the different terms in the augmented Lagrangian functional have the same amount of differentiation for function $u$. The augmented Lagrangian functional is defined as

$$L_r(q, u, \lambda) = \frac{1}{2} \| u - u_d \|_{L^2(\Omega)}^2 + \frac{1}{2} \gamma \| \nabla u - \bar{u}_g \|_{L^2(\Omega)}^2 + \beta R(q) + \frac{r}{2} \| -\nabla \cdot (q \nabla u) - f \|_{H^{-1}(\Omega)}^2 + (\lambda, -\nabla \cdot (q \nabla u) - f)_{H^{-1}(\Omega)}$$

for $q \in K, \quad u \in H^1_0(\Omega), \quad \lambda \in H^{-1}(\Omega)$.

Based on (2.2), (2.3), (2.4), (2.5), and (2.6), it is easy to see that

$$L_r(q, u, \lambda) = \frac{1}{2} \| u - u_d \|_{L^2(\Omega)}^2 + \frac{1}{2} \gamma \| \nabla u - \bar{u}_g \|_{L^2(\Omega)}^2 + \beta R(q)$$

$$+ \frac{r}{2} (C(A_qu - f), A_qu - f) + (\lambda, C(A_qu - f))$$

$$= \frac{1}{2} \| u - u_d \|_{L^2(\Omega)}^2 + \frac{1}{2} \gamma \| \nabla u - \bar{u}_g \|_{L^2(\Omega)}^2 + \beta R(q)$$

$$+ \frac{r}{2} (C(B_uq - f), B_uq - f) + (\lambda, C(B_uq - f)).$$

Dropping the boundary terms, one gets

$$\frac{\partial L_r}{\partial q} = \beta R'(q) + rB_u^*C(B_uq - f)$$

$$+ B_u^*C\lambda,$$

$$\frac{\partial L_r}{\partial u} = u - u_d - \gamma \nabla \cdot (\nabla u - \bar{u}_g),$$

$$+ rA_u^*C(A_qu - f)$$

$$+ A_u^*C\lambda,$$

$$\frac{\partial L_r}{\partial \lambda} = -\nabla \cdot (q \nabla u) - f.$$
It is known that the saddle point for $L_r$ satisfies the optimality conditions
\begin{equation}
\left( \frac{\partial L_r}{\partial q} , \phi - q \right) \geq 0 \quad \forall \phi \in K,
\end{equation}
\begin{equation}
\frac{\partial L_r}{\partial u} = 0,
\end{equation}
\begin{equation}
\frac{\partial L_r}{\partial \lambda} = 0.
\end{equation}

We shall use an iterative procedure similar to that for (P1) to find the saddle point.

### 3.3. Augmented Lagrangian for (P3)

Similar to (P2), we shall use the same $H^{-1}$-norm to enforce the equation constraint. The augmented Lagrangian functional is now
\[
L_r(q, u, \lambda) = \frac{1}{2} \| u - u_d \|_{L^2(\Omega)}^2 + \frac{1}{2} \gamma \| q \nabla u - \vec{u}_v \|_{L^2(\Omega)}^2 + \beta R(q)
\]
\[
+ \frac{r}{2} \| - \nabla \cdot (q \nabla u) - f \|_{H^{-1}(\Omega)}^2 + (\lambda, - \nabla \cdot (q \nabla u) - f)_{H^{-1}(\Omega)}
\]
for $q \in K, u \in H^1_0(\Omega), \lambda \in H^{-1}(\Omega)$.

In a similar way as for (P2), we can calculate that
\[
\frac{\partial L_r}{\partial q} = \gamma (q \nabla u - \vec{u}_v) \cdot \nabla u + \beta R'(q)
\]
\[
+ r B_u^* C (B_u q - f)
\]
\[
+ B_u^* C \lambda,
\]
\[
\frac{\partial L_r}{\partial u} = u - u_d - \gamma \nabla \cdot (q^2 \nabla u - q \vec{u}_v)
\]
\[
+ r A_u^* C (A_u u - f)
\]
\[
+ A_u^* C \lambda.
\]

In the discrete setting, the variational equation for $\frac{\partial L_r}{\partial q} = 0$ is
\[
\gamma (q \nabla u - \vec{u}_v, \phi \nabla u) + \beta (R'(q), \phi)
\]
\[
+ r (C(B_u q - f), B_u \phi)
\]
\[
+ (C \lambda, B_u \phi) = 0 \quad \forall \phi \in P_h.
\]

The algorithm to search for a saddle point for $L_r$ is given later.

### 4. Noise removal preprocessing

In many practical applications, observations contain random noise. Let $u$ be a function and $u_d$ be its observation with random noise. From the observation $u_d$, we try to recover a more accurate function for $u$. In order to preserve the shape of the function and at the same time filter out the highly oscillatory noise, we use a TV denoising procedure. More precisely, we assume that the observation $u_d$ and the true function $u$ satisfy
\[
\| u - u_d \|_{L^2(\Omega)} = \sigma,
\]
where $\sigma$ is the $L^2$-noise level which we assume to be known a priori. Numerical evidence shows that it is not necessary to know $\sigma$ exactly. From the observation $u_d$, we shall solve
\[
\min_{v \in W^{1,1}_0(\Omega)} TV(v) \quad \text{under constraint} \quad \frac{1}{2} \| u - u_d \|_{L^2(\Omega)}^2 = \frac{1}{2} \sigma^2.
\]
The solution of this minimization problem has the smallest TV-norm and at the same time satisfies the noise level constraint. In practice, we replace $TV(v)$ by its approximation $R(v)$. To solve this problem, we use a simple Lagrangian multiplier iterative procedure which is a simple extension of the projection algorithm used in Rudin, Osher, and Fatemi [41]. Taking $u_0 = u_d$ and choosing $\lambda_0 > 0$ large (for example, $\lambda_0 = 10^9$), we do

$$
-\nabla \cdot \left( \frac{\nabla u_n}{\sqrt{(|\nabla u_n-1|)^2 + \varepsilon}} \right) + \lambda_{n-1}(u_n - u_d) = 0 \quad \text{in } \Omega , \quad u = 0 \quad \text{on } \partial \Omega ,
$$

$$
\lambda_n = -\left( \frac{\nabla u_n}{\sqrt{(|\nabla u_n|)^2 + \varepsilon}} \nabla(u_n - u_d) \right)/\sigma^2 .
$$

The formula for updating $\lambda_n$ is formally obtained by multiplying both sides of the first equation of (4.1) by $u_n - u_d$ and using the noise level constraint. Note that the newly computed $u_n$ is used to replace $u_n - 1$ in the updating of $\lambda_n$. The above iterative procedure only needs to solve a simple linear equation. For more efficient numerical methods for solving this problem, we refer to [12, 11].

**4.1. Noise removing for (P1).** Before using Algorithm 1 to solve the inverse problem (P1), we shall use the above noise removing technique for $u_d$ to get a denoised solution $u$ of (4.1). Then, we use $u$ as the observation data and also as the initial value for $u_0$ in Algorithm 1.

As the solution $u$ of (1.1) cannot have discontinuities, the $H^1$-norm could be a natural choice for the denoising. In our simulations, both the TV-norm and the $H^1$-norm denoising have been used. In most of the cases, the two approaches do not differ very much. However, there are examples where the TV-norm works slightly better due to the nonlinear diffusion effect; i.e., the TV-norm denoising is adding very little diffusion at the places that $|\nabla u|$ is very large. In using the TV-norm denoising, the choice of $\varepsilon$ affects the accuracy of the smoothed function. If $\varepsilon$ is chosen too small, then the recovered function is very flat near the points that $\nabla u = 0$. In order to avoid this problem, it is not desirable to choose $\varepsilon$ very small. In the simulations, we use two different values for $\varepsilon$ for $R(\cdot)$ to approximate the TV-norm. The value of $\varepsilon$ for solving (4.1) is taken as $\varepsilon = 0.1$. The value of $\varepsilon$ for the regularization of the coefficient is taken as $\varepsilon = 0.01$ or smaller.

**4.2. Noise removing for (P2).** As observations for both $u$ and $\nabla u$ are available, we shall solve

$$
\min_{v \in H^1_0(\Omega)} \left( \frac{1}{2} \|v - u_d\|^2_{L^2(\Omega)} + \frac{1}{2} \omega \|\nabla v - \bar{u}_g\|^2_{L^2(\Omega)} \right)
$$

to get a denoised $u$. The choice of the parameter $\omega$ is not very important. In all our experiments, we use $\omega = 1$. Minimization (4.2) is equivalent to solving

$$
\left\{ 
\begin{array}{ll}
-\omega \nabla \cdot (\nabla u - \bar{u}_g) + u - u_d = 0 & \text{in } \Omega , \\
    u = 0 & \text{on } \partial \Omega .
\end{array}
\right.
$$

Due to the large amount of diffusion introduced by the operator $\Delta = \nabla \cdot \nabla$, it is not necessary to introduce extra TV-norm or $H^1$-norm regularization, which is equivalent to introducing extra artificial diffusion into the denoising procedure. After solving (4.3), we solve (4.1) for each component of $\bar{u}_g$ and this removes the noise from $\bar{u}_g$ directly.
4.3. Noise removing for (P3). For the inverse problem (P3), we shall again solve (4.1) to remove the noise from $u_d$. To remove the noise from $\vec{u}_v$ we observe that

$$-\nabla \cdot (q \nabla u) = f \quad \text{and} \quad q \nabla u \approx \vec{u}_v.$$  

Therefore, we shall find a minimizer for

$$\min_{\vec{w} \in W^{1,1}(\Omega)} \alpha TV(\vec{w}) + \frac{1}{2} \| \vec{w} - \vec{u}_v \|_{L^2(\Omega)}^2 + \frac{1}{2} \omega \| \nabla \cdot \vec{w} + f \|_{L^2(\Omega)}^2,$$

which removes the noise from $\vec{u}_v$ and at the same time enforces the equation constraint. When the noise level is known, we could use the Lagrangian multiplier method similar to the one used for (4.1) to solve

$$\min_{\vec{w} \in W^{1,1}(\Omega)} TV(\vec{w}) + \frac{1}{2} \omega \| \nabla \cdot \vec{w} + f \|_{L^2(\Omega)}^2 \quad \text{under constraint} \quad \frac{1}{2} \| \vec{w} - \vec{u}_v \|_{L^2(\Omega)}^2 = \sigma^2.$$  

In our calculations, the parameter $\omega$ is taken to be 1.

5. Iterative schemes for the augmented Lagrangian problems.

5.1. Augmented Lagrangian for (P1). We shall use the following algorithm to search for a saddle point for $L_r$ over $K \times H^1_0(\Omega) \times H^1_0(\Omega)$ in an iterative way. The main idea is to alternately solve the minimization problem in one of the two variables $u$ and $q$ assuming the other is known. A few steps of this alternating minimization procedure is then followed by an updating step on the Lagrange multiplier $\lambda$.

Algorithm 1.

Step 1. Choose $u_0 \in H^1_0(\Omega), \lambda_0 \in H^1_0(\Omega)$; and $r > 0$.
Step 2. Set $u_n^0 = u_{n-1}$. For $k = 1, 2, \ldots, k_{\text{max}}$, do:

Step 2.1. Solve $q_n^k = \arg \min_{q \in K} L_r(q, u_{n-1}^{k-1}, \lambda_{n-1}),$ which gives [23]

$$\begin{align}
(5.1) \quad \beta R'(q_n^k) + r B_{u_n^{k-1}}^* C^* C (B_{u_n^{k-1}} - q_n^k - f) \\
+ B_{u_n^{k-1}}^* C^* \lambda_{n-1} - \phi - q_n^k) \geq 0 \quad \forall \phi \in K.
\end{align}$$

Step 2.2. Solve $u_n^k = \arg \min_{u \in H^1_0(\Omega)} L_r(q_n^k, u, \lambda_{n-1}),$ which gives

$$\begin{align}
(5.2) \quad u_n^k - u_d + r A_{q_n^k}^* C^* C (A_{q_n^k} u_n^k - f) \\
+ A_{q_n^k}^* C^* \lambda_{n-1} = 0.
\end{align}$$

Step 3. Set $u_n = u_n^k, q_n = q_n^k,$ and update $\lambda_n$ as

$$\begin{align}
(5.3) \quad \lambda_n = \lambda_{n-1} + re(q_n, u_n).
\end{align}$$

If we use finite element approximations in our computations, $A_{q_n^k}$ is a matrix that depends on $q_n^k$, and $B_{u_n^{k-1}}$ is a matrix that depends on $u_n^{k-1}$. If the finite element mesh is quasi uniform, we can use inverse inequalities to show that the TV-norm is equivalent to the Sobolev norms $H^k(\Omega), \ k = 1, 2$. The equivalent constant depends on the mesh size $h$. Making use of this equivalence, we can employ the same techniques as in [35].
to show that there is a unique saddle point for $L_r$ if the observation error is sufficiently small, i.e., $\|\tilde{u} - u_d\| << 1$. Moreover, the iterative solution of Algorithm 1 converges to the saddle point with a linear convergence rate. However, if the observation error is large, we can only show as in [13] that a subsequence of \{(q_n, u_n, \lambda_n)\} converges to a saddle point.

It is known that it is not necessary to choose the penalization parameter $r$ very large for the augmented Lagrange method. In our calculations, we take

$$r = \frac{\|u_d\|_{L^2(\Omega)}}{\|Cf\|_{L^2(\Omega)}} \times 10$$

to balance the least-square term $\frac{1}{2}\|u - u_d\|^2$ and the penalization term $\frac{r}{2}\|e(q, u)\|^2$.

We shall use $u = 0$ for the boundary condition of (5.2). The variational form of (5.1) implies a Neumann boundary condition for $q^k_n$. In our simulations, we use a simple projection method to handle the variational inequality (5.1); i.e., we first find $q^k_n$ which solves

$$\beta R'(q^k_n) + r B^*_{u_{n-1}} C^* C (B_{u_{n-1}} q^k_n - f) + B^*_{u_{n-1}} C^* \lambda_{n-1} - q^k_n = 0.$$  (5.4)

The above equation comes from the variational form

$$\beta (R(q^k_n), \phi) + r \left( C (B_{u_{n-1}} q^k_n - f), C B_{u_{n-1}} \phi \right) + (\lambda_{n-1}, C B_{u_{n-1}} \phi) = 0 \quad \forall \phi \in P_h.$$  (5.5)

Due to the use of the TV-norm regularization, the matrix $R'(q)$ is nonlinear with respect to $q$, and we use a technique similar to [49] to deal with this nonlinearity. After solving the nonlinear equation (5.4), we set

$$q^k_n = \max(k_1, \min(q^k_n, k_2));$$  (5.6)

i.e., we project $q^k_n$ into the convex subset $K$. If we iterate between (5.4) and (5.6), the obtained $q^k_n$ will converge to the minimizer of (5.1). In our simulations, we compute (5.4) and (5.6) only once. The constant $k_1 > 0$ is taken to be very small and $k_2$ is taken to be very large. In most of our tests, the solution of (5.4) is in the interior of the admissible coefficient set $K$. This same technique is used to solve the variational inequalities of the algorithms for (P2) and (P3).

5.2. Augmented Lagrangian for (P2). The algorithm used to find the saddle point for (P2) is similar to that of (P1).

Algorithm 2.

Step 1. Choose $u_0 \in H^1_0(\Omega), \lambda_0 \in H^{-1}(\Omega)$, and $r > 0$.

Step 2. Set $u^0_n = u_{n-1}$. For $k = 1, 2, \ldots, k_{\text{max}}$, do:

Step 2.1. Solve for $q^k_n \in K$ from

$$\left( \beta R'(q^k_n) + r B^*_{u_{n-1}} C (B_{u_{n-1}} q^k_n - f) + B^*_{u_{n-1}} C \lambda_{n-1}, \phi - q^k_n \right) \geq 0 \quad \forall \phi \in K.$$  (5.7)
Step 2.2. Solve for $u^k_n \in H^1_0(\Omega)$ from
\begin{align*}
u^k_n - u_d - \gamma \nabla \cdot (\nabla u^k_n - \bar{u}_g) \\
+ r A^*_{q_n} C (A_{q_n} u^k_n - f) \\
+ A^*_{q_n} C \lambda_{n-1} &= 0.
\end{align*}
(5.8)

Step 3. Set $u_n = u^k_n$, $q_n = q^k_n$, and update the multiplier as
$$\lambda_n = \lambda_{n-1} + r (A_{q_n} u_n - f).$$

In order to balance the contribution of the different terms, the parameters $\gamma$ and $r$ are chosen according to
$$\gamma = \frac{\|u_d\|_{L^2(\Omega)}}{\|\nabla \cdot \bar{u}_g\|_{L^2(\Omega)}} \times 100, \quad r = \frac{\|u_d\|_{L^2(\Omega)}}{\|f\|_{L^2(\Omega)}} \times 10.$$

The extra factors 100 and 10 used for $\gamma$ and $r$, respectively, are added based on our numerical experiences. Choosing $\gamma$ and $r$ in this way, the iterative procedure has a faster convergence.

A homogeneous Dirichlet boundary condition is used for (5.8). The boundary condition of (5.7) is implicitly contained in the variational form. Similar to the solution of (5.1), we first get $q^k_n$ from the nonlinear equation
\begin{align*}
\beta R'(q) + r B^*_{u_n} C (B_u u_n - f) \\
+ B^*_{u_n} C \lambda_{n-1} = 0
\end{align*}
(5.9)
and then project $q^k_n$ into the set $K$ as described in (5.6) for (P1). The matrix corresponding to $R'(q)$ depends on $q$. We use a technique similar to that of [49] to solve the nonlinear equation.

By assuming the observation error is sufficiently small, we can prove that there exists a unique saddle point for $L_r$ and the iterative solution of Algorithm 2 converges to the saddle of $L_r$; see [35]. If the observation errors are large, we can only prove that there is a convergent subsequence; see [13].

5.3. Augmented Lagrangian for (P3). We use the following analogous algorithm to search for a saddle point for $L_r$.

ALGORITHM 3.

Step 1. Choose $u_0 \in H^1_0(\Omega), \lambda_0 \in H^{-1}(\Omega)$ and $r > 0$.

Step 2. Set $u_0^n = u_{n-1}$. For $k = 1, 2, \ldots , k_{\text{max}}$, do:

Step 2.1. Solve for $q^k_n \in K$ from
$$\left( \gamma (q^k_n \nabla u^{k-1}_n - \bar{u}_g) \cdot \nabla u^{k-1}_n + \beta R'(q^k_n) \\
+ r B^*_{u_n} C (B_u u_n - f) \\
+ B^*_{u_n} C \lambda_{n-1}, \phi - q^k_n \right) \geq 0 \quad \forall \phi \in K.$$

Step 2.2. Solve for $u^k_n \in H^1_0(\Omega)$ from
$$\left( \nu^k_n - u_d - \gamma \nabla \cdot (\nabla u^k_n - \bar{u}_g) \\
+ r A^*_{q_n} C (A_{q_n} u^k_n - f) \\
+ A^*_{q_n} C \lambda_{n-1} = 0. \right.$$
Step 3. Set \( u_n = u_n^k, q_n = q_n^k \), and update the multiplier as

\[
\lambda_n = \lambda_{n-1} + r(A_{q_n} u_n - f).
\]

The variational inequality for \( q_n^k \) is solved by using the simple project procedure as described in (5.4)–(5.6). The parameters \( \gamma \) and \( r \) are determined by

\[
\gamma = \frac{\| u_d \|_{L^2(\Omega)}}{\| \nabla \cdot u_d \|_{L^2(\Omega)}}, \quad r = \frac{\| u_d \|_{L^2(\Omega)}}{\| f \|_{L^2(\Omega)}} \times 10.
\]

**6. Numerical experiments.** We first treat the discretization issues. Let \( \Omega \subset R^n, n = 1, 2, 3 \), be a bounded domain. We first divide \( \Omega \) into finite elements \( T_h = \{ e_i \} \). In all the simulations, a uniform mesh is used. The domain is discretized by simplicial elements, namely intervals, triangles, and tetrahedrons in one, two, and three dimensions, respectively. Let \( S_h \) denote the piecewise linear finite element space over \( T_h \) with zero Dirichlet boundary value on \( \partial \Omega \). Let \( P_h \) denote the piecewise constant finite element space over \( T_h \). The space \( S_h \) will be used to approximate \( u \) and the space \( P_h \) will be used to approximate \( q \). Let \( e_i \) and \( e_j \) be any two elements of the finite element division \( T_h \) and \( |\bar{e}_i \cap \bar{e}_j| \) be the \((n-1)\)-dimensional measure of the interface between \( \bar{e}_i \) and \( \bar{e}_j \). For a given \( q \in P_h \), we define \( q_i \) to be \( q|_{e_i} \). Then it is easy to calculate that

\[
TV(q) = \int_\Omega |\nabla q| dx = \sum_{i<j} |q_i - q_j||\bar{e}_i \cap \bar{e}_j|.
\]

Correspondingly, we define the discrete functional \( R(q) \) as

\[
R(q) = \sum_{i<j} \left( \sqrt{\frac{|q_i - q_j|^2}{h^2} + \varepsilon} \right) |\bar{e}_i \cap \bar{e}_j| \cdot h.
\]

It is clear that \( R(q) \to TV(q) \) when \( \varepsilon \to 0 \). With \( R(q) \) thus defined, it is easy to form the matrix \( R'(q) \). The functions from \( S_h \) are \( H^1 \)-functions and \( R(\cdot) \) given in (2.1) is already well defined for \( S_h \) and also satisfies \( R(q) \to TV(q) \) as \( \varepsilon \to 0 \). The observations for our simulations are generated as follows. Let \( f(x) \) be a given function. For any given \( q \in P_h \), let \( u \in S_h \) be the solution of

\[
(q\nabla u, \nabla v)_\Omega = (f, v)_\Omega \forall v \in S_h.
\]

The observation \( u_d \) is obtained from

\[
u_d = u + \delta_1 \ast R_1 \ast \| u \|_{L^2(\Omega)},
\]

where \( \delta_1 \) is a constant controlling the noise level and \( R_1 \) is a vector with \( \dim(R_1) = \dim(u_d) \) of uniformly distributed random numbers in \([-1, 1]\) with zero mean. The observations \( \bar{u}_g \) and \( \bar{u}_v \) are generated similarly by adding random noise to their true finite element values, i.e.,

\[
\bar{u}_g = \nabla u + \delta_2 \ast R_2 \ast \| \nabla u \|_{L^2(\Omega)},
\]

\[
\bar{u}_v = q\nabla u + \delta_3 \ast R_3 \ast \| q\nabla u \|_{L^2(\Omega)}.
\]
TV REGULARIZATION FOR ELLIPTIC INVERSE PROBLEMS

Our implementation is in MATLAB with a machine precision of about $10^{-15}$. The random vectors $R_i$ are generated by the MATLAB function `rand.m`. In all our experiments, we take $\delta_1 = \delta_2 = \delta_3 = \delta$.

In identifying the coefficient $q(x)$, Algorithms 1 and 2 for (P1) and (P2) are sensitive to the value of the regularization parameter $\beta$. This is true especially when the observation errors are large. In our simulations, we use a trial-and-error approach to adjust the value of $\beta$. In using Algorithms 1–3, it is not necessary to do many inner iterations between Step 2.1 and Step 2.2; we use only three iterations (i.e., $k_{\max} = 3$). The initial value of the Lagrange multiplier $\lambda$ is always taken to be $\lambda_0 = 0$.

Numerical tests given later show that the noise removal preprocessing procedure introduced in section 4 is important for getting an accurate estimated coefficient. This can be explained by the dependence of matrix $B$ on the gradient $\nabla u$; see (2.7). The matrix $B$ is needed in solving the equations for $q$ in the algorithms. We know that the convergence of Algorithms 1–3 depends on the fact that we can have a good initial value for $u_0$, i.e., $u_0 = u_d$. This is observed in our numerical simulations and also can be explained by the analysis of Tai and Kunisch [35, section 3]. When $u_d$ contains large observation errors and the mesh size is very small, $\nabla u_d$ is a very oscillatory function with large variations in function value and can destroy the convergence of the algorithms. The noise removal preprocessing procedure can filter out the oscillatory noise and remove the nonphysical large variations in the function values.

6.1. One-dimensional experiments. In the experiments, we try to identify a one-dimensional piecewise continuous $q$ with several jump discontinuities from different kinds of observations by using the formulations (P1), (P2), and (P3), respectively. We take $f(x) = 1$. The true coefficient $q(x)$ is chosen as (see Figure 1)

$$q(x) = \begin{cases} 
0.5, & x \in [0, 1/5] \cup [2/5, 3/5] \cup [4/5, 1], \\
2, & x \in [1/5, 2/5] \cup [3/5, 4/5].
\end{cases}$$
The interval $[0, 1]$ is divided into 200 elements, i.e., $h = 1/200$.

6.1.1. The algorithm for (P1). First, we use Algorithm 1 to identify the coefficient from an $L^2$-observation. The identified $q(x)$ for different noise levels are shown in Figure 1. Different values for $\beta$ are used for each subfigure. We solve (4.1) to smooth the observation $u_d$. Then we use the smoothed $u_d$ as the observation and also as the initial value for $q$ in Algorithm 1. Since the TV functional $R(q)$ is nonlinear in $q$, we need an initial guess for $q$ for the iterative solution of the equation in Step 2.1. The results in Figure 1 are computed with the initial value $q = 1$. Our experience is that the algorithm is very robust with respect to the choice of the initial value for $q$.

In Figure 2, we show the identified coefficients from the observations by using the original observation data without preprocessing. All the parameters used by the algorithm, i.e., $\beta, \gamma, r, \epsilon$, are kept unchanged compared to the ones used for Figure 1. We can clearly see that the identified coefficients are much less accurate than the identified coefficients from the smoothed observation data. We could use large values for the regularization parameter $\beta$, and the identified coefficient from the original noise observation data will be less oscillatory. For the same amount of noise, the original noise observation data always needs much larger regularization value to kill the oscillations, and the identified coefficient is less accurate due to the extra smoothing effect.

6.1.2. The algorithm for (P2). We identify the same coefficient $q(x)$ by observing both the value of $u$ and the gradient of $u$ using formulation (P2). The identified $q$ is shown in Figure 3.

If the boundary condition was known exactly, we performed tests that showed the observation for $u$ was not necessary, i.e., we could get the same accuracy for the identified coefficient only from the observation of the gradient of $u$. 
The identified coefficient with different noise levels by $H^1$-observation. $\kappa = \delta \| R^2 \|_{L^2(\Omega)} / \| \nabla u \|_{L^2(\Omega)}$. 

In the simulation, we have observed that the noise removal preprocessing procedure not only removes most of the random noise but also preserves the shape of the true state $u$; see Figure 4. In Figure 5, the identified coefficients from the original noisy observation data without preprocessing are shown. The parameters $\beta, \gamma, r,$ and
\[ \kappa = 0.10976\%, \quad \delta = 0.1\% \quad \beta = 1 \times 10^{-13} \]

\[ \kappa = 1.0976\%, \quad \delta = 1\% \quad \beta = 1 \times 10^{-11} \]

\[ \kappa = 5.4882\%, \quad \delta = 5\% \quad \beta = 1 \times 10^{-9} \]

\[ \kappa = 10.9765\%, \quad \delta = 10\% \quad \beta = 2 \times 10^{-9} \]

\[ \kappa = 32.9295\%, \quad \delta = 30\% \quad \beta = 1 \times 10^{-8} \]

\[ \kappa = 54.8824\%, \quad \delta = 50\% \quad \beta = 1 \times 10^{-8} \]

\[ \text{Fig. 5. The identified coefficients with different noise levels by } H^1\text{-observation from the original noisy observation data.} \]

\[ \epsilon \] use the same values as Figure 3. It is clear that the identified coefficients from the smoothed data are better than the ones from the original noisy observation data.

### 6.1.3. The algorithm for (P3)

We now present results for formulation (P3); i.e., we identify \( q(x) \) by observing both \( u \) and \( q \nabla u \). The identified \( q \) is shown in Figure 6. Note that different values of \( \beta \) are used for different observation errors; cf. Figure 6.

The performance does not seem to be as good as for formulation (P2). In particular, in order to identify the location of the discontinuities of \( q(x) \) with an accuracy of 1%, we need the observation error to be less than 1%. The identified coefficients from the original noise observations are shown in Figure 7.

### 6.1.4. Summary for the one-dimensional tests

Summarizing our numerical experience, we can see that the observation for the gradient \( \nabla u \) is preferable in situations in which the observation error is large. When the observation errors are small, all three formulations (P1)–(P3) can give accurate solutions. For a given level of observation errors, the identified coefficient of (P2) is more accurate than that of the other two formulations.

It shall also be pointed out that we can construct examples for which (P1) can tolerate as large observation errors as (P2) and the identified coefficient of Algorithm 1 is as accurate as that of Algorithm 2 for large observation errors. It is also easy to construct examples such that the gradient of \( u \) is very large in some small parts of \( \Omega \) but nearly zero in the other parts of \( \Omega \). For such examples, the maximum tolerable observation errors for the algorithms can be much less than the errors for the test problems given above.

When the observation errors are large, the computational error for the identified coefficient is dominated by the errors from the observation and the regularization parameter \( \beta \) and it is hard to observe the advantages of the TV-norm regularization.
For small observation errors, the advantage of the TV-norm regularization is fully demonstrated by the better accuracy over the $H^1$-norm regularization. In Figure 8, we show the identified coefficients by observations of $u$ and $u_x$ without noise using the TV-norm and the $H^1$-norm regularizations, respectively. The TV-norm enables
us to identify the coefficient with an accuracy of $\|q_h - q\|_{L^2(\Omega)} \approx 10^{-6}$. However, the accuracy of the identified coefficient by the $H^1$-norm regularization is only $\|q_h - q\|_{L^2(\Omega)} \approx 10^{-2}$. The equation errors are plotted for both of the identified coefficients; see Figure 8. In the figure, the equation error $Au - f$ is evaluated from the identified $q_h$ and $u_h$. From the plots, we see that both identified coefficients satisfy the equation constraint to an accuracy of order $10^{-11}$. In Figure 8, we have used $h = 1/200, \delta = 0, \epsilon = 10^{-4}, \beta = 10^{-18}$.

In order to show the influence of the regularization parameter $\beta$, we try to identify
TV REGULARIZATION FOR ELLIPTIC INVERSE PROBLEMS

Figure 10. Figure (a) shows the identified coefficient from \( u(x) + \delta R_1 \) and from \( u_h + \delta R_1 \) and the true coefficient, respectively. No observation is added. The identified coefficient from \( u_h \) is the same as the true coefficient. The identified coefficient from \( u(x) \) contains errors in the region \((0,0.5)\) and near the point that \( u_x = 0 \). Figure (b) shows the identified coefficient from \( u(x) + \delta R_1 \) and from \( u_h + \delta R_1 \) and the true coefficient, respectively. The value of \( \delta \) is 0.01. The identified coefficient from \( u_h + \delta R_1 \) and the identified coefficient from \( u(x) + \delta R_1 \) are nearly the same. Figure (c) shows the approximation error \( u(x) - u_h \) and the added noise \( \delta * R_1 \) with \( \delta = 0.01 \).

A coefficient by using different values for \( \beta \). The identified coefficients by different values of \( \beta \) are given in Figure 9. We assume that the observations for \( u \) are available with 1% of observation errors, i.e., \( \delta = 10^{-2} \). From this test, it is clear that large or small \( \beta \) is not preferable in the computation.

The particular way that the observation data are generated is sometimes called an "inverse crime"; i.e., the same method that simulates the forward problem in the output-least-squares code is used to generate the data. The advantage of getting data in this way is that we can easily test our algorithms for different \( f \) and \( q \) functions. In order to show that our method also works for more realistic observation data, we have chosen a problem with an analytical solution. The observation data are generated by adding random noise to the nodal values of the analytical solution. In such a case, the observation data contains two kinds of noise: one is the approximation error and another is the random noise. Denote by \( u(x) \) the continuous analytical solution and by \( u_h \) the corresponding finite element solution for \( u(x) \). In Figure 10, we plot the identified coefficients from both \( u(x) + \delta R_1 \) and from \( u_h + \delta R_1 \). When the added noise is much less than the approximation error (cf. Figures 10(a) and 10(c)), the identified coefficient from \( u_h + \delta R_1 \) is better than the identified coefficient from \( u(x) + \delta R_1 \). In the case where the added noise is as large or much larger than the approximation error, the identified coefficients from both observations are nearly the same; cf. Figure 10(b).

6.2. Two-dimensional experiments. In two dimensions, the computational cost is increased dramatically and the efficient solution of the algebraic equations becomes a more critical issue. Therefore, we only present a few examples with large observation errors. We will not go into the details of the numerical procedures in the present paper. We refer to Kunisch and Tai [34] and Tai et al. [44] for some algorithms.
that can solve the algebraic equations of Algorithms 1–3 more efficiently. In [34], a nonoverlapping domain decomposition is combined with the augmented Lagrangian method. In [44], overlapping domain decomposition and multigrid methods are used as preconditioners to solve the algebraic equations of Algorithms 1–3.

First, we use Algorithm 1 to identify a piecewise smooth coefficient from an observation of \( u \) with random errors using formulation (P1). The true coefficient is 
\[
q(x, y) = c_1(x, y),
\]
where \( c_1(x, y) \) is a discontinuous piecewise constant function with values \( c_1 = 2 \) or \( c_1 = 1 \); see Figure 11 for the location of the discontinuity. Similar to the one-dimensional problems, the maximum tolerable observation error for \( L^2 \)-observations is less than 1%. In Figures 11 and 12, the subfigure in the left corner on the top shows the solution \( u(q) \) corresponding to the true coefficient \( q \) we want to recover. The observation \( u_d \) is produced by adding random noise to \( u(q) \) and the added noise is shown in the middle on the top. Before we recover the coefficient, we use the noise removal algorithms of section 4 to smooth the noise. After the smoothing, the noise is shown in the subfigure in the right corner on the top, which is clearly less oscillatory. The true \( q \) and the recovered \( q, u \) are shown in the bottom row of the figures. The observation error for Figure 11 is \( \delta = 1\% \). In the computation, we have used \( h = 1/128, r = 100, \beta = 0.00125, \epsilon = 0.001 \). In order to show the influence of the value of \( \beta \) on the accuracy of the identified coefficient, we also present another computational result for identifying the same coefficient. In Figure 12, we have used \( \beta = 0.00625 \). It can be seen that the identified coefficient is less oscillatory and the “tower” in the middle of the figure of the identified coefficient is lower.

In Figure 13, we identify the same \( q \) by observations of \( u \) and \( \nabla u \) using formulation (P2). A 50% observation error is added, i.e., \( \delta = 0.5 = 50\% \). In the left column of the figure, the true coefficient and the corresponding \( u, u_x, u_y \) are plotted. The recovered \( q \) and \( u \) are shown in the subfigures on the top. The other subfigures show the noise which was used to produce the observations and the actual noise after using the smoothing algorithm of section 4. The true \( u_x \) and \( u_y \) have jumps at the discontinuities of \( q \). After adding a 50% noise, it is impossible to detect the jump in the observations for \( \nabla u \). The noise smoothing algorithm of section 4 is

\[ u \]

\[ \text{Original random noise} \]

\[ \text{Noise after smoothing} \]

\[ \text{True coef.} \]

\[ \text{Identified coef.} \]

\[ \text{Identified u} \]

Fig. 11. The identified coefficient by observing \( u \) with \( \delta = 1\% \) and \( \beta = 0.00125 \).
especially powerful when we have observations for both $u$ and $\nabla u$. The smoothed noise is less oscillatory and also must less in amplitude. The maximum tolerable observation error is about 50%. The constants used in the computation for Figure 13 are $h = 1/128$, $r = 100$, $\beta = 0.00625$, $\epsilon = 0.001$.

Numerical tests using observations of $u$ and $q\nabla u$ by formulation $(P3)$ have also been done. The maximum tolerable observation error is about 1%. The conclusions that we can draw from the two-dimensional experiments are similar to those of the one-dimensional tests, namely that formulation $(P2)$ can tolerate more observation errors than $(P1)$ and $(P3)$. For a given level of observation error, the identified coefficient of $(P2)$ is more accurate than that of the other two formulations.

7. Conclusion. From the numerical tests, it can be seen that the preprocessing procedure is useful when the observation errors are large. The smoothing procedure can remove the random noise and produce a much more accurate observation. If we do not use the smoothing procedure, it has been observed that a finer mesh tolerates a smaller noise error. For a given noise level, the accuracy of the identified coefficient deteriorates as the mesh size decreases. Such a phenomenon does not appear if we use the smoothing for the observation data, especially if we have the gradient observation.

When the coefficient is discontinuous, it is better to use the TV-norm regularization. When the observation errors are large, the identified coefficient using the $H^1$- or $H^2$-regularization can often be as good as the coefficient using the TV-norm regularization. The advantages of the TV-norm regularization are more visible for smaller observation errors.

The inverse coefficient estimation problem has been traditionally addressed using observation of the state $u$. Our numerical tests show that the observation of the gradient $\nabla u$ can improve the accuracy of the identified coefficient and can tolerate much larger observation errors. However, the observation of the flux (also called velocity) $q\nabla u$ does not give a much better identified coefficient.

The augmented Lagrangian approach reduces the inverse problem into that of solving some algebraic equations. Efficient methods can be used to solve these alge-
braic equations. This enables us to solve two- and three-dimensional inverse problems with a large number of unknowns; see also [34, 44].

The choice of the values for $\beta$ and $r$ affects the accuracy of the identified coefficient. More studies are needed to find algorithms that can determine the optimal values of $\beta$ and $r$ automatically.

In the parameterization, the coefficient $q$ is approximated by piecewise constants and the state variable $u$ is approximated by piecewise linear functions. If we use piecewise linear finite element functions to approximate both, then it is much more difficult to control the oscillations with large observation errors; i.e., the identified
coefficients are more sensitive to the choices of $\beta$ and $r$ and the noise level.

**Acknowledgments.** The authors thank for J. Frøyen for his help with the two-dimensional numerical experiments. The authors would like to acknowledge the comments by the referees and Uri M. Ascher which led to many improvements in the presentation of this article.

**REFERENCES**