Inverse Problems **23** (2007) 1759–1786 doi:10.1088/0266-5611/23/5/001

A multilevel model correction method for parameter identification

Jingzhi Li and Jun Zou

Department of Mathematics, The Chinese University of Hong Kong, Shatin, NT, Hong Kong, Peoples Republic of China

E-mail: jzli@math.cuhk.edu.hk and zou@math.cuhk.edu.hk

Received 13 September 2006, in final form 14 March 2007
Published 22 August 2007
Online at stacks.iop.org/IP/23/1759

Abstract
Output least-squares formulation with Tikhonov regularization is one of the most frequently used and reliable methods for parameter identification in PDEs. As the discretized optimization system of the formulation is nonlinear, ill-conditioned, and constrained with some PDEs, its numerical solution is very time-consuming and often unstable. In this paper, we present a multilevel model correction method, which aims at solving the general nonlinear output least-squares optimization system in an efficient and robust manner. The multilevel method is conducted iteratively so that larger part of its computations is done on a sequence of nested coarse meshes defined on the concerned physical domain. The key ingredient of the method is to update the output least-squares objective functional on each coarse mesh by using the most updated information of the identifying parameter from a finer mesh. Numerical experiments are provided to demonstrate the robustness and efficiency of the method.

(Some figures in this article are in colour only in the electronic version)

1. Introduction

Parameter identification in PDEs arises widely in scientific and engineering applications. For example, in a heat conductive system, it may be necessary to reconstruct different model parameters, such as diffusivity, radiativity, source term, heat flux and initial temperature, from the measurement of temperature in a subregion of the physical domain within a certain time interval. For more details about the physical background of parameter identification problems, we refer to [3, 6, 8, 9, 12, 20, 25] and the references therein.

Due to the nonlinear and ill-posed nature of parameter identification problems, the output least-squares formulation with Tikhonov regularization has been regarded as one of the most effective and reliable approaches. The least-squares formulation transforms an originally ill-posed parameter identification problem into a regularized optimization system. However, the resulting optimization systems are constrained with some PDEs, and are often highly nonlinear and numerically ill-conditioned. How to solve these nonlinear optimization systems efficiently remains challenging. Most existing algorithms involve the alternative solutions of a direct PDE and its adjoint at each iteration, namely the direct solver computes the model state with the current estimated parameter while the adjoint solver seeks an adjoint variable based on the previous model state to update the identifying parameter. Thus the computational complexity of parameter identification increases rapidly with the size of the discrete system, and a practical numerical identification method should be able to significantly reduce the complexity of the identification process.

Numerical solutions of the output least-squares systems with Tikhonov regularization have drawn increasing attention in the last two decades, and various identification algorithms have been proposed and implemented. In principle, these methods can be categorized into the following classes:

(1) Gradient-type methods [17, 18], which solve the concerned PDE and its adjoint equation to get the gradient direction for the identifying parameter and update the parameter iteratively along the gradient direction.
(2) The nonlinear conjugate gradient method [8, 22, 24], which solves the concerned PDE and its adjoint equation to get the gradient direction for the parameter, then modifies this gradient to get its conjugate gradient by some nonlinear conjugate gradient schemes, and updates the parameter iteratively along the conjugate gradient direction.
(3) The Newton or quasi-Newton method, such as the L-BFGS method [10, 14, 23], which approximates the Hessian of the nonlinear functional based on its gradient knowledge.
(4) The Lagrangian or augmented Lagrangian method [3, 6, 13, 18], which introduces a Lagrangian or augmented Lagrangian functional, and then updates iteratively the parameter, the solution of the PDE system and the Lagrangian multiplier simultaneously or alternatively.

In spite of the rich literature on numerical methods for inverse problems in PDEs, fast and robust solvers are rare. All the traditional methods mentioned in the four classes above are conducted at only one level of mesh, which limits the feasibility of the methods in the case of large-scale discrete systems. Taking into consideration just one solution for the direct and adjoint PDE at each iteration, the computational cost of the entire identifying process can already be formidable in large-scale applications.

In this paper, we propose a *multilevel model correction* (MMC) method for solving the output least-squares Tikhonov regularized optimization systems based on the finite element discretization. As will be seen, the MMC method is efficient and robust in terms of computational complexity for numerical identification of model parameters in PDEs; see some typical model problems in section 3. MMC is a nonlinear multigrid-type algorithm, and it aims at reducing the computational cost greatly by taking advantage of solving the concerned PDE and its adjoint on a sequence of nested coarser meshes, and ensures convergence in the finite element framework by updating the model functional on each coarse mesh by adding a model correction term, using the most updated information of the identifying parameter from a finer mesh. For updating the current estimated parameter on each coarse mesh, a gradient-type method with inexact line search using a simple backtracking rule is used, which will be shown numerically to have a good smoothing property in section 5.

Given a nested sequence of finite element spaces associated with a sequence of nested meshes on the physical domain, we can formulate an output least-squares minimization model on each mesh. It must be emphasized that directly applying the multigrid idea to these models may not lead to the convergence of the resulting method since different regularization parameters are required on different levels of meshes in order to achieve reasonable approximation of the parameter, but the choice of regularization parameters is difficult itself. To avoid choosing regularization parameters at different levels of meshes, a crucial model correction term is constructed at each level, recursively from the finest mesh to the coarsest. The main idea behind this correction term is to implicitly adjust the regularization term in the cost functional at each coarse mesh such that the restriction of the exact parameter at the finer mesh onto the current coarse one is just the exact parameter of the cost functional on the coarse grid. Then we employ the nonlinear multigrid idea to accelerate the identifying process by computing corrections to the parameter through solving those modified coarse grid models. In this manner, most computations of the MMC method are done on coarser meshes, where the PDE-constrained minimization problem can be solved more efficiently because the computational cost for the solution of the direct and adjoint PDEs is much cheaper than that at the finest mesh. Through the information exchange between the coarse and fine meshes, the convergence of the algorithm speeds up tremendously. As one will see, the MMC algorithm can be applied to various identification problems defined on general physical domains. In fact, MMC can be viewed as a general framework to construct fast solvers for general parameter identifications.

Nonlinear multigrid methods originate from the *full approximation scheme* (FAS) method, which was initiated by Brandt from his ingenious dual viewpoint in his seminal paper [5] and used to solve efficiently some forward well-posed problems such as nonlinear elliptic equations. The main idea of multigrid methods is to accelerate the convergence speed of the basic relaxation iteration by computing corrections to the solution on coarser grids, which speeds up the computation significantly. In the last three decades, motivated by the need for fast and efficient algorithms for large-scale problems, nonlinear multigrid methods have been widely used for the numerical solution of nonlinear PDEs.

Owing to the computational advantages of multigrid methods, a lot of efforts have been made in recent years to apply multigrid methods for solving inverse problems in one way or another; see, e.g., [1, 2, 4, 15, 16, 19, 21, 26]. One popular way was to first derive the KKT optimal condition for the constrained minimization problem, then apply the multigrid methods to the KKT optimality system or its linearized system. But for the optimality system, it is hard to construct effective smoothing algorithms as well as to select reasonable regularization parameters on all the coarse grids [1, 2, 4, 15, 16, 19]. An effective smoother was proposed in [19] for some linear inverse problems, then extended to some nonlinear inverse problems in [15, 16]. Elegant convergence and regularization theories were also established for such smoothers [15, 16, 19]. But in order to achieve good smoothing effects, one may have to do some careful estimates and tunings on certain parameters, which can be very hard in applications. Different from this approach, Yamamoto and Zou [26] were the first to propose a multilevel type method using multigrid ideas to solve the least-squares Tikhonov regularized minimization system directly, for which the classical gradient or nonlinear conjugate gradient method turns out to be effective smoothers. More recently, Lewis and Nash [21] extended Brandt's FAS method in the optimization setting and proposed an optimization-based multigrid method for solving optimal control problems associated with PDEs.

Motivated by the dual viewpoint from [5], the idea of multigrid methods for parameter identifications based on the finite element discretization in [26], and the optimization-based multigrid approach in [21], we shall propose a nonlinear multigrid-like method for solving the output least-squares Tikhonov regularized minimization systems directly in this paper, namely the MMC method introduced earlier. We will formulate the method in a general and systematic manner so that readers can easily apply it to many other inverse problems. Compared with [26], a main difference is to incorporate some appropriate model correction terms into the cost functionals at different levels of grids based on the finite element discretization. We point out that it is much more tricky and technical to formulate appropriate model correction terms in the finite element spaces than in the ordinary Euclidean spaces [21].

The rest of the paper is organized as follows. In section 2 we first give an abstract formulation for general parameter identification problems, then introduce an abstract MMC algorithm and prove an important property for the algorithm. In section 3 some typical model problems will be presented. In section 4, we will formulate the PDE-constrained minimization systems, describe the finite element discretization, and derive the gradient formulae for the cost functionals for some typical parameter identifications in elliptic and parabolic systems. Numerical experiments are presented to illustrate the efficiency and robustness of the new approach in section 5.

2. General framework of the MMC method

2.1. Abstract framework of parameter identifications

Consider a general parameter identification problem which is described by the relation, often a partial differential equation,

$$\mathcal{L}(q, u) = 0 \tag{2.1}$$

where $\mathcal{L}$ is a partial differential operator, depending on a parameter q from a parameter space Q and the solution u in a state space V; see section 4 for some typical models. We will consider both V and Q to be Hilbert spaces with inner products $(\cdot, \cdot)_V$ and $(\cdot, \cdot)_Q$, respectively. Let $K \subset Q$ be a closed convex subset of Q and play the role of an admissible parameter set. The parameter identification problem can be both space- and time-dependent.

The objective of a parameter identification procedure is to choose a parameter $q^* \in K$ such that the solution u of (2.1) associated with q^* well matches the measured state. In general, the measured data for the state u may be corrupted by measurement errors. The noisy data with noise level δ will be denoted by z^δ. In practice, we may only measure the state u or its gradient in a small observable subregion ω of Ω. For the time-dependent model, measurement may be further restricted within a certain time period I, say $I = [t_1, t_2]$. Let $Z_\omega, Z_{\omega,I}$ be the measurement sets for the time-independent and -dependent models, or more precisely the restrictions of some Hilbert spaces on ω and $\omega \times I$, respectively. When no confusion is caused, we shall drop the subscript and simply write Z, instead of Z_ω or $Z_{\omega,I}$. Then we see that $z^\delta \in Z$.

Parameter identifications are in general an ill-posed problem. It might have no solutions or have multiple solutions. Even if existence and uniqueness are assured, the parameter might not depend continuously on the measured data, i.e., slight changes in the observed data may lead to significant changes in the parameter. Considering the inherent measurement errors in the observation process, this absence of continuous dependence is quite undesirable from a practical point of view.

To transform the ill-posed parameter identification problem into a nearby stabilized one and make a numerical solution feasible, we shall consider the most frequently used and most reliable methodology which formulates the parameter identification problem into the output least-squares system along with some Tikhonov regularization:

$$\min_{q \in K} J(q) = \frac{1}{2}\|u(q) - z^\delta\|_Z^2 + \gamma N(q), \tag{2.2}$$

where $u = u(q) : K \to V$ solves equation (2.1), $N(q) : Q \to R^1$ is a regularization term and γ is a regularization parameter. For later use, we introduce another Hilbert space $\tilde{Q}$, $Q \subset \tilde{Q}$, which has a similar algebraic structure to that of Q except for a weaker norm and will be used in the model correction step.

Assume that $u(q)$ and $N(q)$ are Gateaux differentiable, and $u'(q)p$ and $N'(q)p$ are the derivatives of $u(q)$ and $N(q)$ at q in direction $p \in Q$, respectively. Then the Gateaux derivative of J at q in direction p can be written as

$$J'(q)p = (u(q) - z^\delta, u'(q)p)_Z + \gamma N'(q)p \qquad \forall p \in Q. \tag{2.3}$$

Then a solution q to the minimization problem (2.2) can be characterized as a solution to the first-order variational problem:

$$J'(q)p = 0 \qquad \forall p \in Q. \tag{2.4}$$

To numerically solve the optimization problem (2.2) with constraint (2.1), discretizations of both the state space and the parameter space should be made first. We shall make use of a sequence of nested finite-dimensional spaces to construct an efficient algorithm.

Assume that $\{Q_{h_k}\}_{k=0}^N$, $\{\tilde{Q}_{h_k}\}_{k=0}^N$, $\{Z_{h_k}\}_{k=0}^N$ and $\{V_{h_k}\}_{k=0}^N$ are nested finite dimensional subspaces of Q, $\tilde{Q}$, Z and V, respectively, such that

$$\begin{aligned} &Q_{h_0} \subset Q_{h_1} \cdots \subset Q_{h_N} \equiv Q_h, \qquad &&\tilde{Q}_{h_0} \subset \tilde{Q}_{h_1} \cdots \subset \tilde{Q}_{h_N} \equiv \tilde{Q}_h, \\ &Z_{h_0} \subset Z_{h_1} \cdots \subset Z_{h_N} \equiv Z_h, \qquad &&V_{h_0} \subset V_{h_1} \cdots \subset V_{h_N} \equiv V_h, \end{aligned} \tag{2.5}$$

and $Q_{h_k} \subset \tilde{Q}_{h_k}$ $(0 \leqslant k \leqslant N)$, and $\mathcal{L}_{h_k}$ and N_{h_k} are some approximations of operators L and N based on the spaces Q_{h_k} and V_{h_k}.

Then we can approximate the problem (2.2) and (2.1) at the kth level $(0 \leqslant k \leqslant N)$ as follows:

$$\min_{q_{h_k} \in K_{h_k}} J_{h_k}(q_{h_k}) = \tfrac{1}{2}\|u_{h_k}(q_{h_k}) - z^\delta\|^2_{Z_{h_k}} + \gamma N_{h_k}(q_{h_k}), \tag{2.6}$$

where $K_{h_k} = K \cap Q_{h_k}$, and $u_{h_k} \in V_{h_k}$ solves

$$\mathcal{L}_{h_k}(q_{h_k}, u_{h_k}) = 0. \tag{2.7}$$

Similarly to (2.3), we can obtain the Gateaux derivative of J_{h_k} at q_{h_k} in direction $p_{h_k} \in Q_{h_k}$ as follows:

$$J'_{h_k}(q_{h_k})p_{h_k} = \left(u_{h_k}(q_{h_k}) - z^\delta, u'_{h_k}(q_{h_k})p_{h_k}\right)_{Z_{h_k}} + \gamma N'_{h_k}(q_{h_k})p_{h_k} \qquad \forall p_{h_k} \in Q_{h_k}. \tag{2.8}$$

Let $\phi^i_{h_k} \in Q_{h_k}$ $(1 \leqslant i \leqslant M_k)$ be a set of basis functions of Q_{h_k} $(1 \leqslant k \leqslant N)$, where M_k is the dimension of Q_{h_k}. For convenience, starting from now on, for any Gateaux differentiable functional $F(q_{h_k})$ on Q_{h_k}, we may often denote by $F'(q_{h_k})$ a function in Q_{h_k}, which is a linear combination of the basis functions $\phi^i_{h_k}$ with its coefficients being $F'(q_{h_k})\phi^i_{h_k}$, namely

$$F'(q_{h_k}) \triangleq \sum_{i=1}^{M_k} \left(F'(q_{h_k})\phi^i_{h_k}\right)\phi^i_{h_k}.$$

Next, we introduce two related but different restriction operators. For any two neighboring subspaces Q_{h_k} and $Q_{h_{k-1}}$, the first restriction operator $I^{h_k}_{h_{k-1}} : Q_{h_k} \to Q_{h_{k-1}}$ should have the ability of approximating the fine resolution parameter well at the coarse level and satisfies

$$I^{h_k}_{h_{k-1}} K_{h_k} \subset K_{h_{k-1}}, \tag{2.9}$$

which prevents the restricted fine resolution parameter from going beyond the admissible parameter set at the coarse level. The second restriction operator $\tilde{I}^{h_k}_{h_{k-1}} : Q_{h_k} \to Q_{h_{k-1}}$ is a projection operator satisfying for any $q_{h_k} \in Q_{h_k}$:

$$\left(\tilde{I}^{h_k}_{h_{k-1}} q_{h_k}, \phi_{h_{k-1}}\right)_{\tilde{Q}_{h_{k-1}}} = \left(q_{h_k}, \phi_{h_{k-1}}\right)_{\tilde{Q}_{h_{k-1}}} \qquad \forall \phi_{h_{k-1}} \in Q_{h_{k-1}}. \tag{2.10}$$

The operator is well defined since $Q_{h_k} \subset \tilde{Q}_{h_k}$. Finally, we introduce a prolongation operator $I_{h_k}^{h_{k-1}}$, which will be a natural injection operator from $Q_{h_{k-1}}$ to Q_{h_k} by using their nestedness and has the following property:

$$\left(\tilde{I}_{h_{k-1}}^{h_k}\phi_{h_k}, \phi_{h_{k-1}}\right)_{\tilde{Q}_{h_{k-1}}} = \left(\phi_{h_k}, I_{h_k}^{h_{k-1}}\phi_{h_{k-1}}\right)_{\tilde{Q}_{h_{k-1}}} \qquad \forall\phi_{h_{k-1}} \in Q_{h_{k-1}}, \qquad \phi_{h_k} \in Q_{h_k}. \quad (2.11)$$

These loosely defined operators give us the freedom to devise a number of variants under the MMC framework. Some concrete examples will be given in section 4.

The discretizations in (2.6) and (2.7) usually have a certain stabilizing effect. However, to obtain high quality reconstruction of the model parameter at the finest level ($k = N$), the dimension of the discretized state and parameter spaces grows quickly from coarse to fine grids; thus the discrete problem inherits more and more instability properties from the original infinite dimensional ill-posed problems. These phenomena make the corresponding discrete problem still nonlinear and numerically ill-conditioned and thus involve more computational complexity for solving the problem at the finest level.

To escape the dilemma of either sacrificing the accuracy of the parameter or introducing more computations, we shall solve the discrete problems at different levels in an appropriate way. The crux of the matter is to solve the coarse level problems to recover quickly the profile of the parameter and to leave the details of the parameter to be achieved by the fine level solvers. This is the original idea of [26]. In [26], however, the fixed constant regularization parameters γ at all levels may lead to poor performance in some cases. One should choose different regularization parameters γ_k in (2.6) to obtain a numerically reasonable approximation to the true parameter at different levels, but that is very hard. A more convenient approach is to keep the regularization parameter from the finest level to be used at all other levels, but update the cost functionals at coarser levels by adding an appropriate model correction term such that the restriction of the exact fine grid parameter onto the coarse one is still optimal at that grid. Therefore we can recover the parameter from coarse to fine levels with better and finer resolution while still ensuring the convergence of the algorithm. This is exactly what we are going to do next.

2.2. *Dual viewpoint and model correction*

Traditional linear multigrid methods, which deal with linear systems of equations, are hard to naturally generalize to nonlinear settings. Brandt's *full approximation scheme* (FAS) [5] originating from a dual viewpoint is an alternative approach to directly apply the multigrid idea for solving nonlinear PDEs.

In the design of nonlinear multigrid algorithms for optimization systems, apart from the common ingredients of most multigrid algorithms, two crucial variations are to find a suitable nonlinear smoothing method to smooth the errors and to devise a procedure for approximating correction terms on coarser grids. Comparatively, the latter plays a more fundamental role. The model correction step in our MMC algorithm is devised in light of Brandt's dual viewpoint.

Now suppose that $q_{h_N}^*$ is a minimizer at the Nth level and γ is used as the regularization parameter at all levels; then from (2.8) we have

$$J'_{h_N}(q_{h_N}^*)\phi_{h_N}^i = 0 \qquad \text{for all} \quad \phi_{h_N}^i \in Q_{h_N}\ (1 \leqslant i \leqslant M_N). \quad (2.12)$$

Consider the relative truncation error $\tau_{h_N}^*$, which is a residual function at the coarse level $Q_{h_{N-1}}$ relative to the fine level Q_{h_N}. The coefficient of $\tau_{h_N}^*$ with respect to $p_{h_{N-1}} \in Q_{h_{N-1}}$ is defined as follows:

$$\tau_{h_N}^*(p_{h_{N-1}}) = J'_{h_{N-1}}\left(I_{h_{N-1}}^{h_N} q_{h_N}^*\right)p_{h_{N-1}} - \left(\tilde{I}_{h_{N-1}}^{h_N} J'_{h_N}(q_{h_N}^*), p_{h_{N-1}}\right)_{\tilde{Q}_{h_{N-1}}}. \quad (2.13)$$

From (2.12), it shows that $J'_{h_N}(q^*_{h_N})$ vanishes, then together with the definition of $\tilde{I}^{h_N}_{h_{N-1}}$ we see that the last term in (2.13) is zero. If it is required that the restricted parameter $I^{h_N}_{h_{N-1}} q^*_{h_N}$ is a minimizer at the coarse level, the first-order variational problem $J'_{h_{N-1}}(q_{h_{N-1}})p_{h_{N-1}} = 0$ should be modified on the coarse space $Q_{h_{N-1}}$ as

$$J'_{h_{N-1}}(q_{h_{N-1}})p_{h_{N-1}} - \tau^*_{h_N}(p_{h_{N-1}}) = 0 \qquad \forall p_{h_{N-1}} \in Q_{h_{N-1}}. \tag{2.14}$$

In other words, $\tau^*_{h_N}$ can be regarded as the correction to the first-order variational problem for the original cost functional $J_{h_{N-1}}$ on the coarse grid so that the solution of the coarse level model will be the fine resolution restricted onto the coarse level, i.e., $I^{h_N}_{h_{N-1}} q^*_{h_N} = q^*_{h_{N-1}}$.

Clearly one cannot compute $\tau^*_{h_N}$ without knowledge of $q^*_{h_N}$, but we do have an approximation to $q^*_{h_N}$, namely the current estimated parameter q_{h_N}. Using this in equation (2.13), we get an approximation of $\tau^*_{h_N}(p_{h_{N-1}})$:

$$\tau_{h_N}(p_{h_{N-1}}) = J'_{h_{N-1}}(I^{h_N}_{h_{N-1}} q_{h_N})p_{h_{N-1}} - (\tilde{I}^{h_N}_{h_{N-1}} J'_{h_N}(q_{h_N}), p_{h_{N-1}})_{\tilde{Q}_{h_{N-1}}}. \tag{2.15}$$

Now we are on a crucial step toward our goal. We can construct a modified cost functional $\mathcal{J}_{h_{N-1}}$ at the coarse level by replacing $\tau^*_{h_N}$ with τ_{h_N} in equation (2.14) such that

$$\mathcal{J}'_{h_{N-1}}(q_{h_{N-1}})p_{h_{N-1}} = J'_{h_{N-1}}(q_{h_{N-1}})p_{h_{N-1}} - \tau_{h_N}(p_{h_{N-1}}) \qquad \forall p_{h_{N-1}} \in Q_{h_{N-1}}. \tag{2.16}$$

From relations (2.15) and (2.16), we see that once we obtain the exact solution $q^*_{h_N}$ at the fine level, then $J'_{h_N}(q^*_{h_N})$ will vanish; thus the solution for the modified coarse level model functional $\mathcal{J}_{h_{N-1}}$ is just the restriction of the fine solution on the coarse grid, or $I^{h_N}_{h_{N-1}} q^*_{h_N}$.

To derive the explicit formula of $\mathcal{J}_{h_{N-1}}$ from (2.16), the reverse process of differentiation indicates $\mathcal{J}_{h_{N-1}}$ should be the sum of $J_{h_{N-1}}$ and a linear functional of $q_{h_{N-1}}$ on $\tilde{Q}_{h_{N-1}}$. By the Riesz representation theorem, the functional can be written as

$$-(w_{h_{N-1}}, q_{h_{N-1}})_{\tilde{Q}_{h_{N-1}}}, \tag{2.17}$$

where $w_{h_{N-1}} \in Q_{h_{N-1}}$ is determined in such a way that

$$(w_{h_{N-1}}, p_{h_{N-1}})_{\tilde{Q}_{h_{N-1}}} = \tau_{h_N}(p_{h_{N-1}}) \qquad \forall p_{h_{N-1}} \in Q_{h_{N-1}}. \tag{2.18}$$

Functional (2.17) is just what we called the *model correction term*. Therefore, we can write $\mathcal{J}_{h_{N-1}}$ in the following form:

$$\mathcal{J}_{h_{N-1}}(q_{h_{N-1}}) = \frac{1}{2}\|u_{h_{N-1}}(q_{h_{N-1}}) - z^\delta\|^2_{Z_{h_{N-1}}} + \gamma N_{h_{N-1}}(q_{h_{N-1}}) - (w_{h_{N-1}}, q_{h_{N-1}})_{\tilde{Q}_{h_{N-1}}}. \tag{2.19}$$

For uniform notation, we introduce a dummy function $w_{h_N} = 0 \in Q_{h_N}$ and rewrite J_{h_N}, the same as $\mathcal{J}_{h_N}$, as follows:

$$\mathcal{J}_{h_N}(q_{h_N}) = J_{h_N}(q_{h_N}) = \frac{1}{2}\|u_{h_N}(q_{h_N}) - z^\delta\|^2_{Z_{h_N}} + \gamma N_{h_N}(q_{h_N}) - (w_{h_N}, q_{h_N})_{\tilde{Q}_{h_N}}; \tag{2.20}$$

then the functional $\mathcal{J}_{h_{N-1}}(q_{h_{N-1}})$ in (2.19) changes to

$$\begin{aligned}\mathcal{J}_{h_{N-1}}(q_{h_{N-1}}) = {}& \frac{1}{2}\|u_{h_{N-1}}(q_{h_{N-1}}) - z^\delta\|^2_{Z_{h_{N-1}}} + \gamma N_{h_{N-1}}(q_{h_{N-1}}) \\ & - (\tilde{I}^{h_N}_{h_{N-1}} w_{h_N} + w_{h_{N-1}}, q_{h_{N-1}})_{\tilde{Q}_{h_{N-1}}}.\end{aligned} \tag{2.21}$$

The rest of the work is to generalize the idea to even coarser levels and construct the modified cost functionals $\mathcal{J}_{h_{N-2}}(q_{h_{N-2}}), \ldots, \mathcal{J}_{h_0}(q_{h_0})$. The derivations are almost the same as for deriving (2.21). Suppose we have already obtained the modified cost functional $\mathcal{J}_{h_k}$, which can be written as follows:

$$\mathcal{J}_{h_k}(q_{h_k}) = J_{h_k}(q_{h_k}) = \frac{1}{2}\|u_{h_k}(q_{h_k}) - z^\delta\|^2_{Z_{h_k}} + \gamma N_{h_k}(q_{h_k}) - (w_{h_k}, q_{h_k})_{\tilde{Q}_{h_k}}. \tag{2.22}$$

Then its natural restriction at the next coarser mesh takes the form

$$\tilde{\mathcal{J}}_{h_{k-1}}(q_{h_{k-1}}) = \tfrac{1}{2}\|u_{h_{k-1}}(q_{h_{k-1}}) - z^\delta\|^2_{Z_{h_{k-1}}} + \gamma N_{h_{k-1}}(q_{h_{k-1}}) - (\tilde{I}^{h_k}_{h_{k-1}} w_{h_k}, q_{h_{k-1}})_{\tilde{Q}_{h_{k-1}}}, \tag{2.23}$$

while the final corrected cost functional at the coarse mesh is

$$\mathcal{J}_{h_{k-1}}(q_{h_{k-1}}) = \tfrac{1}{2}\|u_{h_{k-1}}(q_{h_{k-1}}) - z^\delta\|^2_{Z_{h_{k-1}}} + \gamma N_{h_{k-1}}(q_{h_{k-1}}) - (\tilde{I}^{h_k}_{h_{k-1}} w_{h_k} + w_{h_{k-1}}, q_{h_{k-1}})_{\tilde{Q}_{h_{k-1}}}, \tag{2.24}$$

where $w_{h_{k-1}}$ is determined in the same way as for $w_{h_{N-1}}$ in (2.18) using τ_{h_k}. Therefore, the model correction terms will be constructed in such a way recursively for each cost functional at each coarse grid.

2.3. *Abstract multilevel model correction algorithm*

With the above preparations, we are now ready to propose the multilevel model correction method. The main concern of the MMC algorithm is to solve the discrete minimization system (2.6) at the finest space Q_{h_N} by making use of solvers on the auxiliary coarser spaces Q_{h_k} for $0 \leqslant k < N$.

First, we consider a crucial ingredient of the MMC method, its smoothing step. The smoothing should be able to dampen the high frequency components in the error function very fast. We propose a gradient-based optimization method using inexact line search for the discrete smoother. Other types of smoothing methods may also be used, such as nonlinear conjugate gradient methods.

Smoothing algorithm

```
Procedure : Smooth($\mathcal{J}_{h_k}, q_{h_k}, n_k$)
     for  $i = 1$ to $n_k$
         Solve (2.7) for $u_{h_k} := u_{h_k}(q_{h_k})$;
         Compute the functional value $\mathcal{J}_{h_k}(q_{h_k})$;
         Compute the gradient direction $\mathcal{J}'_{h_k}(q_{h_k})$;
         Set initial step length, $\alpha := \alpha_0$;
         Loop:
             $\hat{q}_{h_k} := q_{h_k} - \alpha \mathcal{J}'_{h_k}(q_{h_k})$;
             Solve (2.7) for $\hat{u}_{h_k} := u_{h_k}(\hat{q}_{h_k})$;
             Compute the functional value $\mathcal{J}_{h_k}(\hat{q}_{h_k})$;
             if $\mathcal{J}_{h_k}(\hat{q}_{h_k}) < \mathcal{J}_{h_k}(q_{h_k})$
                 update $q_{h_k} := \hat{q}_{h_k}$;        break;
             else
                 $\alpha := \alpha/2$;
             end if
         end loop
     end for
     return updated $q_{h_k}$;
end Procedure.
```

This smoothing procedure can be explained as follows. We first solve the direct PDE (2.7) to obtain the state variable $u_{h_k}(q_{h_k})$. With knowledge of the state variable, measured data and current iterate of parameter, we can obtain the gradient direction $\mathcal{J}'_{h_k}(q_{h_k})$ by computing the Gateaux gradients of the cost functional along all directions through some explicit formulae, which will depend on different inverse problems. The step size of line search is determined by a simple backtracking criterion, i.e., reducing the step size by a half if the cost functional is not sufficiently decreased or the step size is overemphasized. The initial step size α_0 is chosen such that the magnitude of $\alpha_0 \|\mathcal{J}'_{h_k}(q_{h_k})\|_\infty$ is of the same order as that of $\|q_{h_k}\|_\infty$. This is to make the magnitude of modification of the parameter comparable to that of the current iterate.

Now, we present the abstract multi-level model correction MMC algorithm as follows:

MMC algorithm

```
Procedure : $\mathtt{MMC}(\mathcal{J}_{h_k}, q^{(0)}_{h_k})$
  if $k = 0$,
    1. Coarse solver:
       $q^{(1)}_{h_0} := \mathtt{Smooth}(\mathcal{J}_{h_0}, q^{(0)}_{h_0}, n_0)$;
  else
    2. Pre-smoothing:
       $q^{(1)}_{h_k} := \mathtt{Smooth}(\mathcal{J}_{h_k}, q^{(0)}_{h_k}, n_k)$;
    3. Compute the model correction term:
       Solve (2.7) for $u^{(1)}_{h_k} := u_{h_k}(q^{(1)}_{h_k})$;
       Compute the gradient direction $\mathcal{J}'_{h_k}(q^{(1)}_{h_k})$;
       $q^{(1)}_{h_{k-1}} := I^{h_k}_{h_{k-1}} q^{(1)}_{h_k}$;
       Solve (2.7) for $u^{(1)}_{h_{k-1}} := u_{h_{k-1}}(q^{(1)}_{h_{k-1}})$;
       Compute $\tau_{h_k}(\phi^i_{h_{k-1}}) = \tilde{\mathcal{J}}'_{h_{k-1}}(q^{(1)}_{h_{k-1}})\phi^i_{h_{k-1}} - (\tilde{I}^{h_k}_{h_{k-1}} \mathcal{J}'_{h_k}(q^{(1)}_{h_k}), \phi^i_{h_{k-1}})_{\tilde{Q}_{h_{k-1}}}$
           as in (2.15);
       Construct $\mathcal{J}_{h_{k-1}}$ by computing $w_{h_{k-1}}$ from $\tau_{h_k}$ as in (2.18);
    4. Recursive iteration:
       $q^{(2)}_{h_{k-1}} := \mathtt{MMC}(\mathcal{J}_{h_{k-1}}, q^{(1)}_{h_{k-1}})$;
    5. Coarse grid correction:
       $e_{h_{k-1}} := q^{(2)}_{h_{k-1}} - I^{h_k}_{h_{k-1}} q^{(1)}_{h_k}$;
    6. Line search:
       $q^{(2)}_{h_k} := q^{(1)}_{h_k} + \alpha I^{h_{k-1}}_{h_k} e_{h_{k-1}}$;
    7. Post-smoothing (optional):
       $q^{(3)}_{h_k} := \mathtt{Smooth}(\mathcal{J}_k, q^{(2)}_{h_k}, m_k)$;  return $q^{(3)}_{h_k}$;
End Procedure.
```

Note that the above MMC algorithm works in a recursive way due to the recursive definition of the model correction term. The MMC algorithm can be described in detail as follows.

(1) *Coarsest grid approximate solver*. If it is on the coarsest grid, i.e., $k = 0$, approximately solve the coarsest grid problem

$$\min_{q_{h_0} \in K_{h_0}} \mathcal{J}_{h_0}(q_{h_0}) \qquad \text{s.t. } \mathcal{L}_{h_0}(q_{h_0}, u_{h_0}) = 0. \tag{2.25}$$

directly with initial guess $q_{h_0}^{(0)}$ to get $q_{h_0}^{(1)}$ and return.

(2) *Pre-smoothing*. If it is not on the coarsest grid, i.e., $k \neq 0$, apply the smoothing algorithm to the optimization system on the grid with the initial guess $q_{h_k}^{(0)}$ to get $q_{h_k}^{(1)}$ in n_k iterations.

(3) *Model correction*. First, we compute the gradient directions $\mathcal{J}'_{h_k}(q_{h_k}^{(1)})$ and $\tilde{\mathcal{J}}'_{h_{k-1}}(q_{h_{k-1}}^{(1)})$ at fine and coarse levels, respectively. We have introduced the general principle to recover $w_{h_{k-1}}$ in the previous subsection. Now let us explain in detail the computational aspect. If we expand

$$w_{h_{k-1}} = \sum_{j=1}^{M_{k-1}} c_j \phi_{h_{k-1}}^j$$

in $Q_{h_{k-1}}$, we need to find a way to determine the coefficients c_i $(1 \leqslant i \leqslant M_{k-1})$. From (2.18) and (2.15), we have for each $\phi_{h_{k-1}}^i \in Q_{h_{k-1}}$ that

$$\left(\sum_{j=1}^{M_{k-1}} c_j \phi_{h_{k-1}}^j, \phi_{h_{k-1}}^i \right)_{\tilde{Q}_{h_{k-1}}} = \tilde{\mathcal{J}}'_{h_{k-1}}(q_{h_{k-1}}^{(1)})\phi_{h_{k-1}}^i - \left(\tilde{I}_{h_{k-1}}^{h_k} \mathcal{J}'_{h_k}(q_{h_k}^{(1)}), \phi_{h_{k-1}}^i\right)_{\tilde{Q}_{h_{k-1}}}$$

or in matrix form

$$Wc = b \tag{2.26}$$

where W is an $M_{k-1} \times M_{k-1}$ matrix and $c, b \in R^{M_{k-1}}$, with their entries given by

$$W_{ij} = \left(\phi_{h_{k-1}}^j, \phi_{h_{k-1}}^i\right)_{\tilde{Q}_{h_{k-1}}}, \qquad c = \left(c_1, c_2, \ldots, c_{M_{k-1}}\right)^T,$$
$$b_i = \tilde{\mathcal{J}}'_{h_{k-1}}(q_{h_{k-1}}^{(1)})\phi_{h_{k-1}}^i - \left(\tilde{I}_{h_{k-1}}^{h_k} \mathcal{J}'_{h_k}(q_{h_k}^{(1)}), \phi_{h_{k-1}}^i\right)_{\tilde{Q}_{h_{k-1}}}.$$

Note that there are some techniques for evaluation of these terms. First, the matrix W are symmetric and we can further reduce it into a diagonal matrix by using the mass lumping scheme in the finite element setting. Next, we can compute $\tilde{\mathcal{J}}'_{h_{k-1}}(q_{h_{k-1}}^{(1)})\phi_{h_{k-1}}^i$ through some existing formula because the gradient formulae for $\tilde{\mathcal{J}}_{h_{k-1}}$ and $\mathcal{J}_{h_k}$ are similar in structure, and the formula of $\mathcal{J}_{h_k}$ are supposed to be able to be derived explicitly. From the definition of $\tilde{I}_{h_{k-1}}^{h_k}$ in (2.10), we see that

$$\left(\tilde{I}_{h_{k-1}}^{h_k} \mathcal{J}'_{h_k}(q_{h_k}^{(1)}), \phi_{h_{k-1}}^i\right)_{\tilde{Q}_{h_{k-1}}} = \left(\mathcal{J}'_{h_k}(q_{h_k}^{(1)}), \phi_{h_{k-1}}^i\right)_{\tilde{Q}_{h_{k-1}}}.$$

Since $Q_{h_{k-1}} \subset Q_{h_k}$, we know $\phi_{h_{k-1}}^i \in Q_{h_{k-1}}$ is a linear combination of the basis functions of Q_{h_k}. Hence the computation for $\left(\mathcal{J}'_{h_k}(q_{h_k}^{(1)}), \phi_{h_{k-1}}^i\right)_{\tilde{Q}_{h_{k-1}}}$ is trivial for we have obtained $\mathcal{J}'_{h_k}(q_{h_k}^{(1)})\phi_{h_k}^i$ $(1 \leqslant i \leqslant M_k)$ beforehand, which facilitates the computation of the second term in b_j.

(4) *Multilevel recursion*. Apply the *MMC* algorithm recursively with initial guess $q_{h_{k-1}}^{(1)}$ to the modified coarse grid problem,

$$\min_{q_{h_{k-1}} \in K_{h_{k-1}}} \mathcal{J}_{h_{k-1}}(q_{h_{k-1}}) \qquad \text{s.t. } \mathcal{L}_{h_{k-1}}(q_{h_{k-1}}, u_{h_{k-1}}) = 0, \tag{2.27}$$

to obtain $q_{h_{k-1}}^{(2)}$. Here $\mathcal{J}_{h_{k-1}}(q_{h_{k-1}})$ acts as the role played on the kth level of grid by $\mathcal{J}_{h_k}(q_{h_k})$.

(5) *Coarse level error correction.* Compute

$$e_{h_k} = I^{h_{k-1}}_{h_k}\big(q^{(2)}_{h_{k-1}} - q^{(1)}_{h_{k-1}}\big). \tag{2.28}$$

(6) *Line search.* Perform a line search to obtain

$$q^{(2)}_{h_k} = q^{(1)}_{h_k} + \alpha e_{h_k}. \tag{2.29}$$

We use the simple backtracking criterion to determine the step size as defined by the previous smoothing procedure. There are chances that the updated parameter jumps outside the admissible parameter set, but we can use some safeguard schemes as penalty or barrier to pull back the parameter.

(7) *Post-smoothing (optional).* Apply m_k iterations of the smoothing algorithm to the k-level problem with the current guess $q^{(2)}_{h_k}$ to get $q^{(3)}_{h_k}$.

Due to the general setting of the above MMC method, we can see that it can be applied to solve general inverse problems. First, one may formulate the concerned inverse problem into a constrained minimization problem as (2.2). Then applying some discretization scheme to the original problem yields basic fine and coarse level models as (2.7) and (2.6). With some explicit gradient formulae of the fine and coarse grid cost functionals, we obtain the modified cost functional on each coarse grid. Therefore, our approach provides a systematic way to fit many existing inverse problems into the multilevel model correction algorithm. We will show in section 4 by some examples to describe how to apply our approach to general parameter identification problems in elliptic and parabolic systems when the measurement of state variable or its gradient is available.

Next, we present an important property of the MMC algorithm.

Theorem 2.1. *If the current estimated parameter $q^{i}_{h_N}$ is a critical point of the functional $\mathcal{J}_{h_N}$ over Q_{h_N}, then $q^{i+1}_{h_N}$, generated by one cycle of the MMC algorithm, is still a critical point.*

Proof. Suppose that $q_{h_N} = q^{i}_{h_N}$ is a critical point of the functional $\mathcal{J}_{h_N}$ over the finest space Q_{h_N}; thus $\mathcal{J}'_{h_N}(q_{h_N}) = 0$. Then at the next coarse level, by the definition of the MMC algorithm and a simple calculation, we have

$$\begin{aligned}\mathcal{J}'_{h_{N-1}}\big(I^{h_N}_{h_{N-1}} q_{h_N}\big) p_{h_{N-1}} &= \tilde{\mathcal{J}}'_{h_{N-1}}\big(I^{h_N}_{h_{N-1}} q_{h_N}\big) p_{h_{N-1}} - \big(w_{h_{N-1}}, p_{h_{N-1}}\big)_{\tilde{Q}_{h_{N-1}}}\\ &= \tilde{\mathcal{J}}'_{h_{N-1}}\big(I^{h_N}_{h_{N-1}} q_{h_N}\big) p_{h_{N-1}} - \tilde{\mathcal{J}}'_{h_{N-1}}\big(I^{h_N}_{h_{N-1}} q_{h_N}\big) p_{h_{N-1}} + \big(\tilde{I}^{h_N}_{h_{N-1}} \mathcal{J}'_{h_N}(q_{h_N}), p_{h_{N-1}}\big)_{\tilde{Q}_{h_{N-1}}}\\ &= 0\end{aligned}$$

for all $p_{h_{N-1}} \in Q_{h_{N-1}}$. This indicates that $I^{h_N}_{h_{N-1}} q_{h_N}$ is a critical point for $\mathcal{J}_{h_{N-1}}(q_{h_{N-1}})$ on the coarser space $Q_{h_{N-1}}$. Hence from the error correction step (2.28) in the MMC algorithm, there is no extra error correction introduced, i.e., $e_{h_N} = 0$. By the recursive definition of MMC iterations, MMC will keep q_{h_N} unchanged after one cycle. □

3. Some typical parameter identification problems

In this section we present some typical examples of parameter identification problems in elliptic and parabolic systems. The weak forms of these PDEs will be given, which are needed later in deriving the explicit formulae for evaluating the Gateaux derivatives of the corresponding output least-squares functionals. All of these identification problems can be solved within the framework of the proposed MMC algorithm.

Our aim is to identify the general parameters by observing the state $u(x)$ or $u(x,t)$, in a subregion or within a certain time interval, of a physical process governed by the following elliptic or parabolic PDEs:

$$-\nabla\cdot(a(x)\nabla u(x))+b(x)u(x)=f(x) \qquad \text{in} \quad \Omega, \tag{3.1}$$

$$\frac{\partial}{\partial t}u(x,t)-\nabla\cdot(a(x)\nabla u(x,t))+b(x,t)u(x,t)=f(x,t) \qquad \text{in} \quad \Omega\times(0,T), \tag{3.2}$$

with the homogeneous Dirichlet boundary condition (assumed for the sake of exposition) and the initial value condition $u(x,0)=u_0(x)$ given. Here Ω is an open bounded domain in R^d ($d=1,2$ or 3) with a boundary Γ consisting of two parts Γ_1 and Γ_2, and a, b and f in (3.1)–(3.2) are assumed to ensure that the forward problem is well-posed, e.g., $a, b\in L^\infty(\Omega)$ and $a(x)\geqslant a_0>0, b(x)\geqslant b_0>0$ for $x\in\bar{\Omega}$. We shall use $q(x)$ or $q(x,t)$ to denote the parameter which is to be identified. Below are some typical parameter identification problems:

(1) Identify the diffusivity $q(x)=a(x)$ in (3.1);
(2) Identify the radiativity $q(x)=b(x)$ in (3.1);
(3) Identify the source term $q(x)=f(x)$ in (3.1);
(4) Identify the diffusivity $q(x)=a(x)$ in (3.2);
(5) Identify the radiativity $q(x,t)=b(x,t)$ in (3.2);
(6) Identify the source term $q(x,t)=f(x,t)$ in (3.2);
(7) Identify the flux $q(x)=a(x)\frac{\partial u}{\partial n}(x)$ on Γ_1 in (3.1), with $u=0$ on Γ_2;
(8) Identify the coefficient $q(x)$ in the Robin boundary condition $a(x)\frac{\partial u}{\partial n}(x)+q(x)u(x)=g(x)$ on Γ in (3.1);
(9) Identify the initial temperature $q(x)=u_0(x)$ in (3.2).

To derive the variational forms associated with (3.1)–(3.2), we use $(\cdot,\cdot)$, $\langle\cdot,\cdot\rangle_\Gamma$ and $\langle\cdot,\cdot\rangle_{\Gamma_2}$ to denote the scalar product of space $L^2(\Omega)$, $L^2(\Gamma)$ and $L^2(\Gamma_2)$ respectively, and $H^1_{\Gamma_2}(\Omega)$ to be the subspace of $H^1(\Omega)$ with functions vanishing on Γ_2. Then the weak formulation for (3.1) reads as Find $u\in V$ such that

$$A(q;u,\phi)=F(q;\phi) \qquad \forall\phi\in V, \tag{3.3}$$

while the weak form for (3.2) is Find $u(\cdot,t)\in V$ such that $u(\cdot,0)=u_0$ in Ω and

$$(u_t,\phi)+A(q;u,\phi)=F(q;\phi) \qquad \forall\phi\in V \tag{3.4}$$

for a.e. $t\in(0,T)$, where $A(q;u,v)$ is a bilinear form with respect to u and v, while $F(q;v)$ is a linear functional of v, and V is an appropriate test space; see table 1 for the detailed forms of A, F and V. If we use $u'(q)p$ to denote the Gateaux derivative of $u(q)$ for direction q, then Gateaux derivatives of A and F for any direction p can be computed as follows:

$$A'(q;u,v)p=A'_q(p;u,v)+A(q;u'(q)p,v), \tag{3.5}$$

$$F'(q;v)p=F'_q(p;v), \tag{3.6}$$

where $A'_q(p;u,\phi)$ and $F'_q(p;\phi)$ have explicit forms in terms p since q appears either linearly in A and F or does not appear at all. For example, with problem 1 we have

$$A'(q;u,v)p=(p\nabla u,\nabla v)+(q\nabla u'(q)p,\nabla v)+(bu'(q)p,v), \qquad F'(q;v)p=0.$$

For all these identification problems, one can work out some explicit formulae to evaluate the Gateaux derivatives of the output least-squares Tikhonov regularized functional (see section 4).

Table 1. Bilinear forms, right-hand functionals and test spaces for problems 1–9.

	$A(q; u, v)$	$F(q; v)$	V
1	$(q\nabla u, \nabla v) + (bu, v)$	(f, v)	$H_0^1(\Omega)$
2	$(a\nabla u, \nabla v) + (qu, v)$	(f, v)	$H_0^1(\Omega)$
3	$(a\nabla u, \nabla v) + (bu, v)$	(q, v)	$H_0^1(\Omega)$
4	$(q\nabla u, \nabla v) + (bu, v)$	(f, v)	$H_0^1(\Omega)$
5	$(a\nabla u, \nabla v) + (qu, v)$	(f, v)	$H_0^1(\Omega)$
6	$(a\nabla u, \nabla v) + (bu, v)$	(q, v)	$H_0^1(\Omega)$
7	$(a\nabla u, \nabla v) + (bu, v)$	$(f, v) + \langle q, v\rangle_{\Gamma_1}$	$H_{\Gamma_2}^1(\Omega)$
8	$(a\nabla u, \nabla v) + (bu, v) + \langle qu, v\rangle_{\Gamma}$	$(f, v) + \langle g, v\rangle_{\Gamma}$	$H^1(\Omega)$
9	$(a\nabla u, v) + (bu, v)$	(f, v)	$H_0^1(\Omega)$

4. MMC algorithm for parameter identifications

We will discuss in this section how to apply the proposed MMC method to the first four parameter identification problems discussed in section 3. The emphasis will be on how to formulate the inverse problems as nonlinear constrained minimization problems, how to yield discretized problems based on finite element discritizations and how to derive the explicit formulae for the Gateaux derivatives of the discrete cost functionals on finite element spaces. All the rest of the work will follow the standard procedure as the abstract MMC algorithm. As for the theoretical details behind this formulation, including the continuity of the discrete cost functional, existence of minimizers and convergence analysis etc, we refer to [6, 17, 24] and the references therein.

4.1. Notation

For the continuous problems, we define the following spaces corresponding to the abstract ones in section 2.1:

$$V = H_0^1(\Omega), \qquad Q = H^1(\Omega), \qquad \tilde{Q} = L^2(\Omega), \qquad Z_\omega = L^2(\omega), \tag{4.1}$$

where Ω is a polyhedral domain in R^d and ω is a subregion of Ω.

Assume that we are given a nested set of shape regular triangulations $\{\mathcal{T}^{h_k}\}_{k=0}^N$ of domain Ω of simplicial elements [7], with $\mathcal{T}^{h_{k+1}}$ being a refinement of $\mathcal{T}^{h_k}$ $(0 \leqslant k \leqslant N-1)$ and ω is always assumed to be a union of some elements of $\mathcal{T}^{h_k}$ $(0 \leqslant k \leqslant N)$, denoted as $\mathcal{T}_\omega^{h_k}$ $(0 \leqslant k \leqslant N)$. $\left\{H_{h_k}\right\}_{k=0}^N$ are continuous piecewise linear finite element spaces defined on $\{\mathcal{T}^{h_k}\}_{k=0}^N$, respectively, such that

$$H_{h_0} \subset H_{h_1} \cdots \subset H_{h_N} \equiv H_h,$$

and the corresponding restriction on ω is denoted by

$$H_{\omega,h_0} \subset H_{\omega,h_1} \cdots \subset H_{\omega,h_N} \equiv H_{\omega,h}.$$

Let $\phi_{h_k}^i$, $i = 1, 2, \ldots, M_k$ be the basis functions on H_{h_k} for $k = 0, 1, \ldots, N$. We denote by $v_{h_k|_\omega}$ the restriction of $v_{h_k} \in H_{h_k}$ onto H_{ω,h_k}, and define the discrete L^2 norm on H_{h_k} or H_{ω,h_k} $(0 \leqslant k \leqslant N)$ by

$$\left\|v_{h_k}\right\|_0^2 = \sum\nolimits_{T\in\mathcal{T}^{h_k}} \left|\left(\bar{v}_{h_k|_T}\right)\right|^2 |T| \qquad \forall v_{h_k} \in H_{h_k}, \tag{4.2}$$

$$\|v_{h_k}\|^2_{0,\omega} = \sum\nolimits_{T\in\mathcal{T}^{h_k}_\omega} \left|\left(\overline{v}_{h_k|_T}\right)\right|^2 |T| \qquad \forall v_{h_k} \in H_{\omega,h_k} \tag{4.3}$$

where $\overline{v}_{h_k|_T}$ is the average of v over the element T and $|T|$ stands for the volume of T.

Next, four discrete spaces corresponding to (2.5) are defined as follows:

$$V_{h_k} = H_{h_k} \cap V, \qquad Q_{h_k} = H_{h_k} \cap Q, \qquad \tilde{Q}_{h_k} = H_{h_k} \cap \tilde{Q}, \qquad Z_{\omega,h_k} = H_{\omega,h_k} \cap Z_\omega. \tag{4.4}$$

To fully discretize the parabolic systems below, we also need the time discretization. To do so, we divide the time interval $(0, T)$ into M equally-spaced subintervals by using nodal points

$$0 = t^0 < t^1 < \cdots < t^M = T$$

with $t^n = n\tau, \tau = T/M$. For a continuous mapping $u : [0, T] \to L^2(\Omega)$, we define $u^n = u(\cdot, n\tau)$ for $0 \leqslant n \leqslant M$. For a given sequence $\{u^n\}_{n=0}^M \subset L^2(\Omega)$ we define the difference quotient and the averaging function

$$\partial_\tau u^n = \frac{u^n - u^{n-1}}{\tau}, \qquad \bar{u}^n = \frac{1}{\tau}\int_{t^{n-1}}^{t^n} u(t)\,\mathrm{d}t. \tag{4.5}$$

The admissible parameter set is given by

$$K = \{q \in Q; \|q\|_Q < \infty \text{ and } 0 < \alpha_1 \leqslant q(x) \leqslant \alpha_2 \text{ a.e. in } \Omega\}, \tag{4.6}$$

while the discrete counterpart for K is given by

$$K_h = \{q_h \in Q_h; 0 < \alpha_1 \leqslant q_h(x) \leqslant \alpha_2 \quad \forall x \in \Omega\}. \tag{4.7}$$

The regularization term can be taken as one of the following terms,

$$N(q) = \int_\Omega q^2\,\mathrm{d}x, \quad \int_\Omega |\nabla q|^2\,\mathrm{d}x, \qquad \text{or} \qquad \int_\Omega \sqrt{|\nabla q|^2 + \varepsilon}\,\mathrm{d}x (\varepsilon > 0), \tag{4.8}$$

which correspond to L^2, H^1 or BV regularization and have the associated Gateaux derivatives in direction $p \in Q$ as follows:

$$N'(q)p = 2\int_\Omega qp\,\mathrm{d}x, \quad 2\int_\Omega \nabla q\cdot\nabla p\,\mathrm{d}x \qquad \text{or} \qquad \int_\Omega \frac{\nabla q\cdot\nabla p}{\sqrt{|\nabla q|^2+\varepsilon}}\,\mathrm{d}x. \tag{4.9}$$

With the above preparations, we are now ready to present the formulations of the general parameter identification problems in elliptic and parabolic systems and derive the explicit formulae for the Gateaux derivatives to be used for the MMC method.

4.2. General parameter identification in elliptic systems

Let z^δ be the observed data in ω, then we can reformulate the first three parameter identification problems (problems 1–3) in section 3 into the following form:

$$\min_{q\in K} J(q) = \tfrac{1}{2}\|u(q) - z^\delta\|^2_{Z_\omega} + \gamma N(q) \tag{4.10}$$

subject to $q \in K$ and $u \equiv u(q) \in V$ satisfying

$$A(q; u, v) = F(q; v) \qquad \forall v \in V. \tag{4.11}$$

We remark that the parameter identifications in problems 7 and 8 can be done basically in a similar manner by replacing the two spaces Q and $\tilde{Q}$ in (4.1) by some other Sobolev spaces defined on the boundary or part of the boundary of Ω.

Let A_h and F_h be some approximations of A and F, then the finite element discretized optimization system can be stated as follows:

$$\min_{q_h \in K_h} J_h(q_h) = \tfrac{1}{2}\|u_h(q_h) - z^\delta\|^2_{Z_{\omega,h}} + \gamma N_h(q_h) \tag{4.12}$$

subject to $q_h \in K_h$ and $u_h \equiv u_h(q_h) \in V_h$ satisfying

$$A_h(q_h; u_h, v_h) = F_h(q_h; v_h) \qquad \forall v_h \in V_h. \tag{4.13}$$

The Gateaux gradient $J_h'(q_h)p_h$ of $J_h(q_h)$ with respect to $p_h \in Q_h$ has the form

$$J_h'(q_h)p_h = (u_h(q_h) - z^\delta, u_h'(q_h)p_h)_{Z_{\omega,h}} + \gamma N_h'(q_h)p_h. \tag{4.14}$$

Note that we can compute $N_h'(q_h)p_h$ according to (4.9) but the computation of $u_h'(q_h)p_h$ is not that easy for the derivative along one direction amounts to one solution of the direct PDE. In order to reduce the cost of evaluating $J_h'(q_h)p_h$, we now introduce an adjoint equation to derive an explicit formula for $J_h'(q_h)p_h$.

Find $w_h \in V_h$ such that

$$A_h(q_h; v_h, w_h) = (u_h(q_h) - z^\delta, v_h)_{Z_{\omega,h}} \qquad \forall v_h \in V_h. \tag{4.15}$$

From (4.13), the Gateaux gradient $u_h'(q_h)p_h$ of $u_h(q_h)$ with respect to p_h satisfies

$$(A_h)'_{q_h}(p_h; u_h, v_h) + A_h(q_h; u_h'(q_h)p_h, v_h) = (F_h)'_{q_h}(p_h; v_h) \qquad \forall v_h \in V_h. \tag{4.16}$$

With the aid of (4.16) and (4.15), we derive the following explicit formula to evaluate the Gateaux gradient $J_h'(q_h)p_h$ of $J_h(q_h)$ with respect to p_h:

$$\begin{aligned} J_h'(q_h)p_h &= (u_h(q_h) - z^\delta, u_h'(q_h)p_h)_{Z_{\omega,h}} + \gamma N_h'(q_h)p_h \\ &= A_h(q_h; u_h'(q)p_h, w_h) + \gamma N_h'(q_h)p_h \\ &= -(A_h)'_{q_h}(p_h; u_h, w_h) + (F_h)'_{q_h}(p_h; w_h) + \gamma N_h'(q_h)p_h. \end{aligned} \tag{4.17}$$

4.3. General parameter identification in parabolic systems

For the parabolic case, let z^δ be the observed data in $\omega \times (t_1, t_2)$, and $q(x)$ be the parameter to be identified; then the identification problems can be reformulated as

$$\min_{q \in K} J(q) = \frac{1}{2}\int_{t_1}^{t_2} \|u(q) - z_\delta\|^2_{Z_\omega}\, \mathrm{d}t + \gamma N(q) \tag{4.18}$$

subject to $q \in K$and $u(\cdot, t) = u(q)(\cdot, t) \in V$ satisfying $u(\cdot, 0) = u_0$ and for a.e. $t \in (0, T)$,

$$(u_t, v) + A(q; u, v) = F(q; v) \qquad \forall v \in V. \tag{4.19}$$

We remark that when the parameter q is both space- and time-dependent, such as $b(x, t)$ and $f(x, t)$ in problems 5 and 6 of section 3, the formulation is basically the same, with only some natural modifications as done in, e.g., [24].

Now using the trapezoidal rule for the time integration and finite element method for space discretization, we obtain the discretized optimization system from (4.18)–(4.19):

$$\min_{q_h \in K_h} J_h(q_h) = \frac{\tau}{2}\sum_{n=n_1}^{n_2} \mu_n \left\|u_h^n - z_\delta^n\right\|^2_{Z_{\omega,h}} + \gamma N_h(q_h) \tag{4.20}$$

subject to $q_h \in K_h$ and $u_h^n \equiv u_h^n(q_h) \in V_h$ satisfying $u_h^0 = P_h u_0$ and for $n = 1, 2, \ldots, M$,

$$\left(\partial_\tau u_h^n, v_h\right) + A_h(q_h; u_h, v_h) = F_h(q_h; v_h) \qquad \forall v_h \in V_h \tag{4.21}$$

where $t_1 = n_1\tau, t_2 = n_2\tau, \mu_{n_1} = \mu_{n_2} = 1/2, \mu_n = 1$ for $n_1 < n < n_2$, and P_h is the L^2-projection from $L^2(\Omega)$ onto V_h (see [17]).

The Gateaux gradient $J_h'(q_h)p_h$ of $J_h(q_h)$ with respect to p_h has the form

$$J_h'(q_h)p_h = \tau\sum_{n=n_1}^{n_2}\mu_n(u_h(q_h) - z^\delta, u_h'(q_h)p_h)_{Z_{\omega,h}} + \gamma N_h'(q_h)p_h. \tag{4.22}$$

Similarly to the elliptic case, we introduce an adjoint equation to reduce the cost of evaluating $J_h'(q_h)p_h$: Find $\{w_h^n\}$ such that $w_h^M = 0, w_h^n \in V_h (0 \leqslant n < M)$ satisfying

$$-\left(\partial_\tau w_h^n, v_h\right) + A_h\left(q_h; v_h, w_h^{n-1}\right) = \theta_n\mu_n(u_h(q_h) - z^\delta, v_h)_{Z_{\omega,h}} \qquad \forall v_h \in V_h \tag{4.23}$$

where $\theta_n = 1$ for $n_1 \leqslant n \leqslant n_2$ and $\theta_n = 0$ otherwise.

Let $\left(u_h^n\right)'(q_h)p_h$ be the Gateaux gradient of u_h^n at q_h in a direction p_h; then we know from (4.21) that $\left(u_h^n\right)'(q_h)p_h \in V_h^0$ satisfies

$$\begin{aligned}&\left(\partial_\tau\left[\left(u_h^n\right)'(q_h)p_h\right], v_h\right) + (A_h)_{q_h}'\left(p_h; u_h^n, v_h\right) + A_h\left(q_h; \left(u_h^n\right)'(q_h)p_h, v_h\right)\\ &\qquad = (F_h)_{q_h}'(p_h; v_h) \qquad \forall v_h \in V_h^0.\end{aligned} \tag{4.24}$$

With the aid of (4.24) and (4.23), we can derive an explicit formula for the Gateaux gradient $J_h'(q_h)p_h$ of $J_h(q_h)$ in direction p_h:

$$\begin{aligned}J_h'(q_h)p_h &= \tau\sum_{n=n_1}^{n_2}\mu_n\left(u_h^n(q_h) - z^\delta, \left(u_h^n\right)'(q_h)p_h\right)_{Z_{\omega,h}} + \gamma N_h'(q_h)p_h\\ &= \tau\sum_{n=1}^{M} -\left(\partial_\tau w_h^n, \left(u_h^n\right)'(q_h)p_h\right) + \tau\sum_{n=1}^{M} A_h\left(q_h; \left(u_h^n\right)'(q_h)p_h, w_h^{n-1}\right)) + \gamma N_h'(q_h)p_h\\ &= \tau\sum_{n=1}^{M}\left(\partial_\tau\left[\left(u_h^n\right)'(q_h)p_h\right], w_h^{n-1}\right) + \tau\sum_{n=1}^{M} A_h\left(q_h; \left(u_h^n\right)'(q_h)p_h, w_h^{n-1}\right)) + \gamma N_h'(q_h)p_h\\ &= -\tau\sum_{n=1}^{M}(A_h)_{q_h}'\left(p_h; u_h^n, w_h^{n-1}\right) + \tau\sum_{n=1}^{M}(F_h)_{q_h}'\left(p_h; w_h^{n-1}\right) + \gamma N_h'(q_h)p_h.\end{aligned} \tag{4.25}$$

Finally, we remark that the formulations and derivations above are almost the same for the general parameter identification problem when the measured data ∇z^δ are available, except that the least-squares term in the cost functionals will be replaced by $\|\nabla u(q) - \nabla z^\delta\|_{Z_\omega}^2$ in the elliptic case and by $\int_{t_1}^{t_2}\|\nabla u(q) - \nabla z^\delta\|_{Z_\omega}^2\,\mathrm{d}t$ in the parabolic case.

5. Numerical experiments

We implement the MMC algorithm using Matlab and test it with a number of parameter identification problems such as the identification of diffusivity, radiativity or source term in Linux Cluster Organon of ITSC at CUHK.

Consider the following elliptic and parabolic equations:

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(q(x)\frac{\mathrm{d}}{\mathrm{d}x}u(x)\right) = f(x), \qquad x \in (0,1), \tag{5.1}$$

$$\frac{\partial}{\partial t}u(x,t) - \frac{\partial}{\partial x}\left(q(x)\frac{\partial}{\partial x}u(x,t)\right) = f(x,t), \qquad (x,t) \in (0,1)\times(0,1) \tag{5.2}$$

$$-\nabla\cdot(q(x,y)\nabla u(x,y)) = f(x,y), \qquad (x,y) \in (0,1)\times(0,1), \tag{5.3}$$

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(a(x)\frac{\mathrm{d}}{\mathrm{d}x}u(x)\right) = q(x), \qquad x \in (0, 1), \tag{5.4}$$

$$-\nabla \cdot (a(x, y)\nabla u(x, y)) + q(x, y)u(x, y) = f(x, y), \qquad (x, y) \in (0, 1) \times (0, 1). \tag{5.5}$$

We are interested in numerical identification of the parameter $q(x)$ or $q(x, y)$ in each equation above. Without loss of generality, the boundary conditions are always assumed to be of homogeneous Dirichlet type and the initial condition will be specified later.

The parameters involved in the MMC algorithm are chosen as follows: the lower and upper bounds α_1 and α_2 in the constraint set K_h are taken to be 0.5 and 20 in (5.1)–(5.3) and (5.5), or −500 and 500 for the source term identification in equation (5.4). The initial guesses q_h^0 for all the test problems are taken to be identically equal to some constant in the entire domain Ω, which is certainly not a good initial guess. We add uniform random noise to the observed data, and the random noise level δ is taken to be 0.01, 0.05 or 0.1. For parabolic cases, the measured data are only given at the terminal time. In the MMC algorithm, we apply two pre-smoothing steps and no post-smoothing steps at each level, and ten optimization steps at the coarsest level. A simple backtracking rule is used for the line search as described in section 2. The errors shown will be the relative L^2-norm error $\|q_h - q\|_0/\|q\|_0$, compared with the exact parameter q.

For one-dimensional problems such as (5.1), (5.2) and (5.4), we divide the entire interval $(0, 1)$ into 160, 80, 40, 20, 10 and 5 equal subintervals respectively, which consist of six levels of nested meshes, with the finest mesh size being $1/160$ while the coarsest mesh size is $1/5$.

For two-dimensional problems such as (5.3) and (5.5), we divide the domain $(0, 1) \times (0, 1)$ into 64×64, 32×32, 16×16, 8×8 and 4×4 equal squares and then further divide each square through its diagonal into two triangles, This leads to five levels of nested meshes over the domain Ω.

The restriction and prolongation operators $I_{h_{k-1}}^{h_k} : Q_{h_k} \to Q_{h_{k-1}}$ and $I_{h_k}^{h_{k-1}} : Q_{h_{k-1}} \to Q_{h_k}$ are taken to be the usual nine-point scheme in two dimensions or the standard three-point weighting schemes in one dimension, and natural prolongation, respectively (see [11]). One can check that $I_{h_{k-1}}^{h_k}$ defined this way satisfies condition (2.9) and that $\tilde{I}_{h_{k-1}}^{h_k}$ and $I_{h_k}^{h_{k-1}}$ are adjoint to each other and satisfy (2.11).

5.1. Smoothing property of gradient methods

Before we carry out our numerical experiments, we first demonstrate how good the smoothing algorithm proposed in section 2.3 is as a smoother.

We show two examples with diffusion parameter identification problems: one is to recover the exact parameter $q_1(x) = 7+2x^2+2\sin(2\pi x)$ in (5.1), the other is to identify the parameter $q_3(x, y) = 7 + 6x^2 y(1 - y)$ in (5.3), using the L^2 observation data without noise. So the cost functional we take is of the form

$$J(q) = \frac{1}{2}\int_\Omega (u(q) - z)^2\,\mathrm{d}x + \gamma \int_\Omega |\nabla q|^2\,\mathrm{d}x.$$

The smoothing performance is basically the same for the other examples.

We deliberately add some high frequency noise to the exact parameters, and then take them as the initial guesses:

$$q_1^{(0)}(x) = q_1(x) + 2\sin(2^i x),$$
$$q_3^{(0)}(x, y) = q_3(x, y) + 2\sin(2^j x)\sin(2^k y).$$

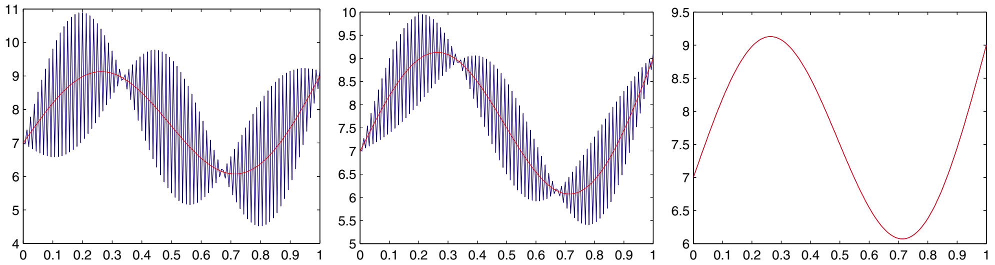


Figure 1. Illustration of the smoothing property of the gradient method in one dimension. From left to right: the initial guess and the first two iterates of the gradient method. The exact parameter is denoted by the smooth red line and the current iterate of the parameter by the background blue line.

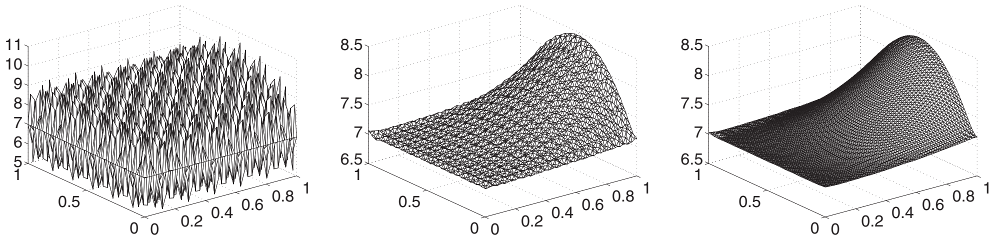


Figure 2. Illustration of the smoothing property of the gradient method in two dimensions. From left to right: the initial guess and the first two iterates of the gradient method.

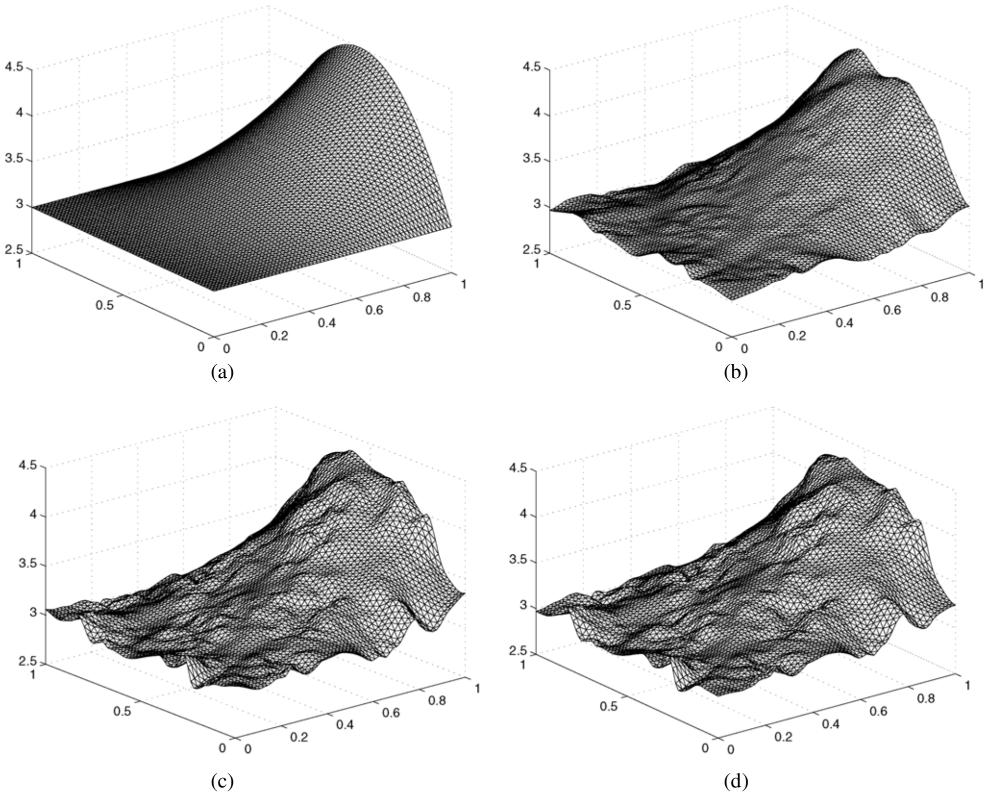
We take $i = 9$, $j = 8$, $k = 8$, and the regularization parameters in the two examples to be $\gamma = 2 \times 10^{-5}$ and $\gamma = 5 \times 10^{-6}$, respectively. Figures 1 and 2 give the initial guess, first and second iterations of the damping process of the high frequency modes respectively by using the smoothing procedure in section 2.3. From the plots, one can observe that the high frequency components can be removed very efficiently by the smoothing algorithm, basically in two iterations.

5.2. Numerical examples and discussions

Example 1. We take the diffusivity $q(x, y)$ and state variable $u(x, y)$ in (5.3) and the observed data ∇z^δ as follows,

$$q(x, y) = 3 + 6x^2 y(1 - y), \qquad u(x, y) = \sin(2\pi x)\sin(2\pi y),$$
$$\nabla z^\delta(x, y) = (1 + \delta R(x, y))\nabla u(x, y),$$

and the function $f(x, y)$ is computed through (5.3) using $u(x, y)$ and $q(x, y)$. Throughout this section, $R(x)$ or $R(x, y)$ will denote a uniform random function varying in the range $[-1, 1]$. Figure 3 gives the exact parameter q_{exact} and the numerically recovered parameter $q_h(x)$, when the observation noise levels δ are 0.01, 0.05 and 0.10 respectively, and a very bad initial guess $q_h^{(0)}$, constant 10 at all mesh points, is used. We can see that the MMC method converges quite fast and satisfactorily even when the random noise is 10%, and the relative error is always less than 5% for all the cases. The profile of the parameter is well approximated



(a)

(b)

(c)

(d)

Figure 3. Identification of diffusivity in example 1: (a) exact solution; (b) $q_h^0 = 10$, $\delta = 0.01$, $\gamma = 5 \times 10^{-4}$, iter = 6, err = 0.0132; (c) $q_h^0 = 10$, $\delta = 0.05$, $\gamma = 1 \times 10^{-3}$, iter = 6, err = 0.0195; (d) $q_h^0 = 10$, $\delta = 0.10$, $\gamma = 2 \times 10^{-3}$, iter = 6, err = 0.0320.

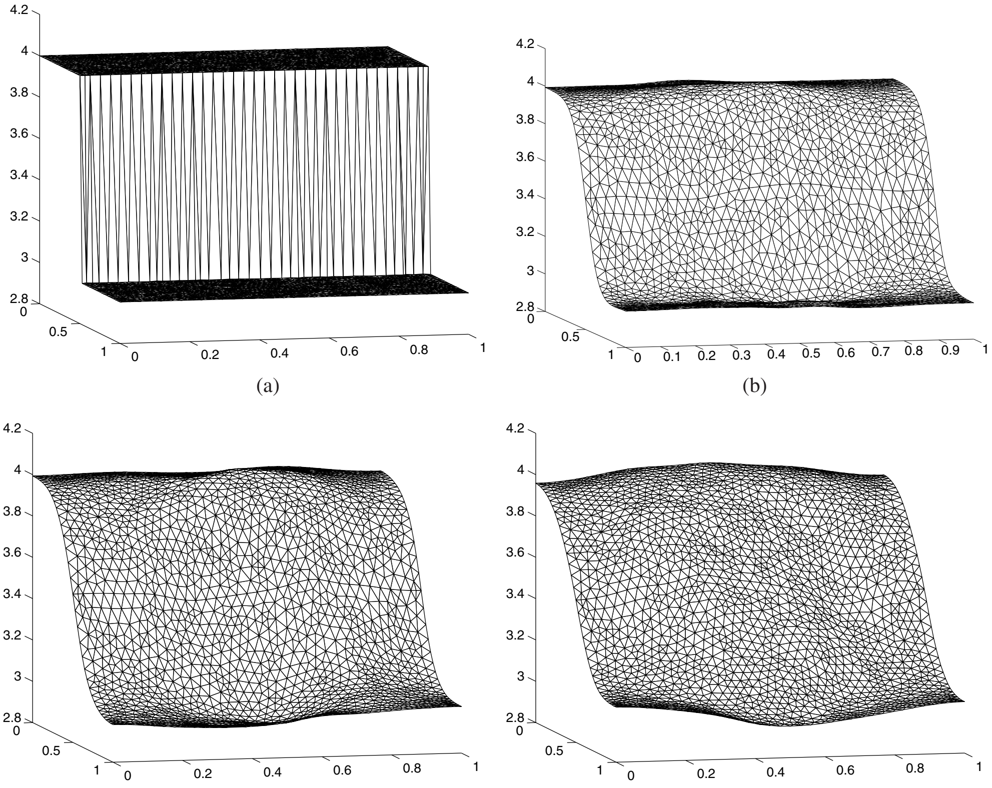
in six iterations, except for those singular points on the boundary or at those points inside the domain where the gradient of u is zero. In fact, these reconstruction results have achieved the best approximation of the output least-squares formulation with Tikhonov regularization, so further iterations cannot improve the reconstruction. This is also true for all the remaining examples presented in this section.

Example 2. We take the discontinuous diffusivity $q(x, y)$ and state variable $u(x, y)$ in (5.3) and the observed data z^δ as follows,

$$q(x, y) = \begin{cases} 4, & x \in (0, 0.5], \quad y \in (0, 1); \\ 3, & x \in (0.5, 1), \quad y \in (0, 1); \end{cases}$$

$$u(x, y) = \sin(\pi x)\sin(\pi y), \qquad z^\delta(x) = (1 + \delta R(x))u(x, y),$$

and the function $f(x, y)$ is computed through (5.3) using $u(x, y)$ and $q(x, y)$. In figure 4, the exact parameter q_{exact} and the identified parameter $q_h(x, y)$ are plotted where the observation noise levels are 0.01, 0.05 and 0.10, respectively. One can see that the numerical reconstruction seems also very satisfactory for this case with discontinuous diffusivity, except that the interface is approximated by a smooth surface with a steep slope, which is natural as



(a)

(b)

(c)

(d)

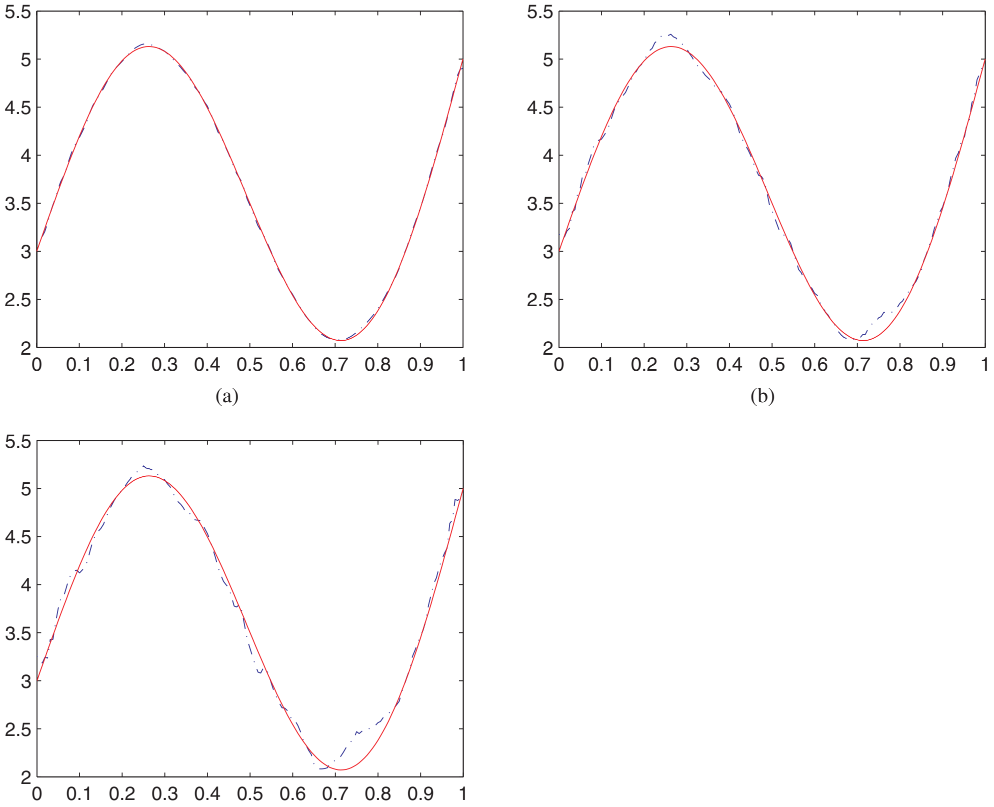
Figure 4. Identification of diffusivity in example 2: (a) exact solution; (b) $q_h^0 = 10, \delta = 0.01, \gamma = 1 \times 10^{-6}$, iter = 19, err = 0.064; (c) $q_h^0 = 10, \delta = 0.05, \gamma = 5 \times 10^{-6}$, iter = 18, err = 0.087; (d) $q_h^0 = 10, \delta = 0.10, \gamma = 1 \times 10^{-5}$, iter = 15, err = 0.132.

the H^1 semi-norm is used as the regularization term. Again the numerical reconstruction is still fast and stable for this tough identification of diffusivity.

Example 3. We take the diffusivity $q(x)$ and state variable $u(x, t)$ in (5.2) and the observed data ∇z^δ at the terminal time as follows,

$$q(x) = 3 + 2x^2 + 2\sin(2\pi x), \qquad u(x, t) = \exp(\sin \pi t)\sin(2\pi x),$$
$$\nabla z^\delta(x) = (1 + \delta R(x))\nabla u(x, 1),$$

the function $f(x, t)$ is computed through (5.2) using $u(x, t)$ and $q(x)$, and the initial boundary condition is given by $u(x, 0) = \sin(2\pi x)$. For this time-dependent problem, the ill-posedness is much more severe. In figure 5, we see that the method converges quite fast and stable for different levels of noises. The profile of the parameter seems to be well approximated in just 8 iterations. Without the speed-up by the MMC scheme, it is extremely hard to achieve similar profiles as in figure 5, even after hundreds of iterations by the standard gradient-type methods.



(a)

(b)

(c)

Figure 5. Identification of diffusivity in example 3: (a) $q_h^0 = 10, \delta = 0.01, \gamma = 8 \times 10^{-7}$, iter $= 8$, err $= 0.015$; (b) $q_h^0 = 10, \delta = 0.05, \gamma = 3 \times 10^{-6}$, iter $= 5$, err $= 0.053$; (c) $q_h^0 = 10, \delta = 0.10, \gamma = 5 \times 10^{-6}$, iter $= 5$, err $= 0.087$.

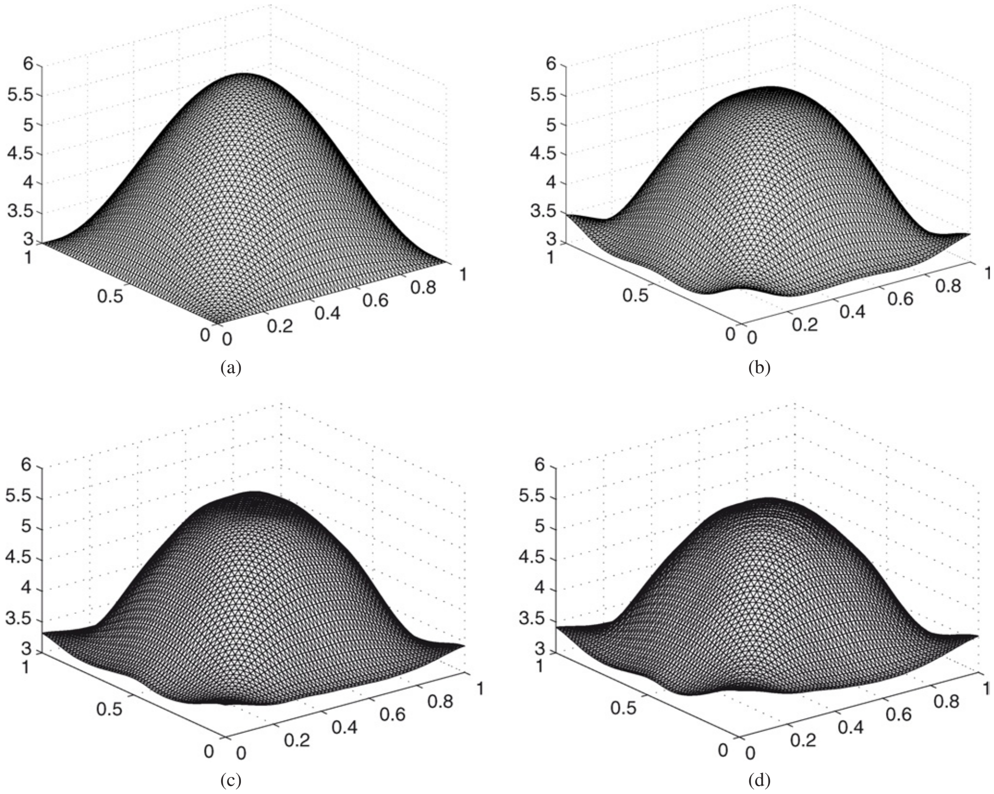
Example 4. We take the diffusivity $q(x, y)$ and state variable $u(x, y)$ in (5.3) and the observed data z^δ as follows,

$$q(x, y) = 3 + 32x(1-x)y(1-y) + \sin(\pi x)\sin(\pi y),$$
$$u(x, y) = \sin(\pi x)\sin(\pi y), \qquad z^\delta(x, y) = (1 + \delta R(x, y))u(x, y),$$

and the function $f(x, y)$ is computed through (5.3) using $u(x, y)$ and $q(x, y)$. Figure 6 presents the exact parameter q_{exact} and the identified parameter $q_h(x)$, with the observation noise levels being 0.01, 0.05 and 0.10, respectively. Although the case with the L^2 observed data is much harder than the case with gradient observation data, the identifying process is still very stable and effective. Hundreds of iterations are needed to achieve similar profiles as in figure 6 if one uses the standard gradient-type methods, as their convergences are extremely slow near the singularities.

Example 5. We take the diffusivity $a(x)$ and state variable $u(x)$ in (5.4) and the observed data ∇z^δ as follows,

$$a(x) = 7 + 2x^2 + 2\sin(2\pi x), \qquad u(x) = \sin(2\pi x), \qquad \nabla z^\delta(x) = (1 + \delta R(x))2\pi\cos(2\pi x),$$



(a) (b) (c) (d)

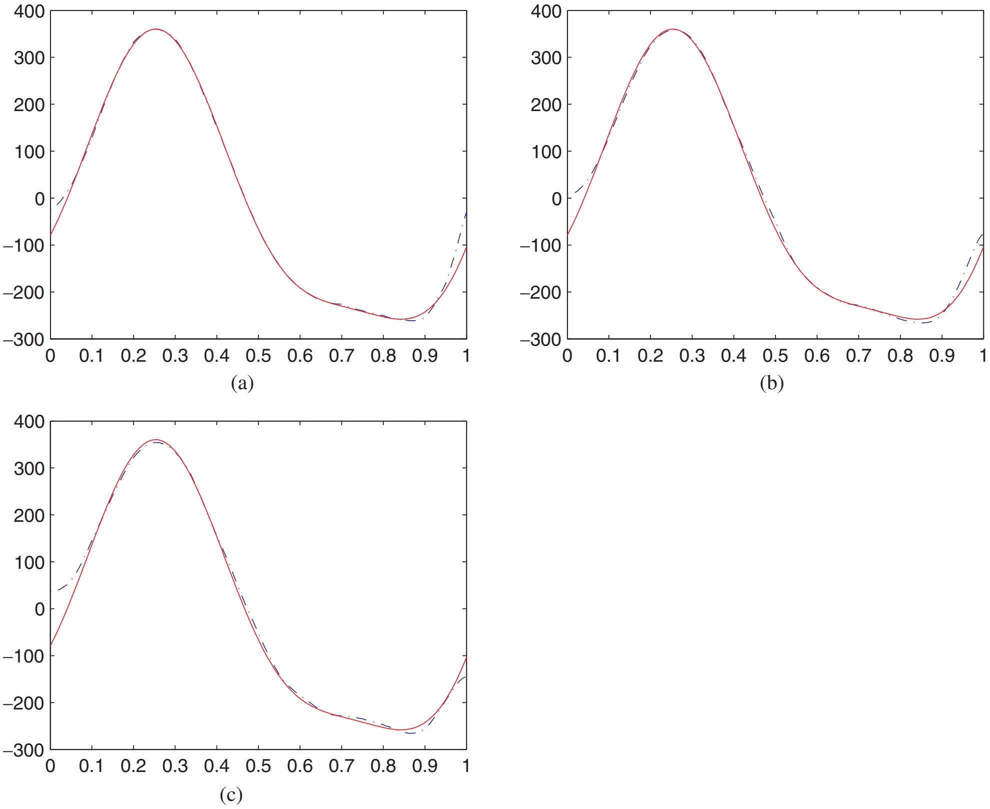
Figure 6. Identification of diffusivity in example 4: (a) exact solution; (b) $q_h^0 = 10$, $\delta = 0.01$, $\gamma = 7 \times 10^{-7}$, iter $= 17$, err $= 0.011$; (c) $q_h^0 = 10$, $\delta = 0.05$, $\gamma = 1 \times 10^{-6}$, iter $= 16$, err $= 0.014$; (d) $q_h^0 = 10$, $\delta = 0.10$, $\gamma = 2 \times 10^{-6}$, iter $= 12$, err $= 0.018$.

and the exact identifying source term $q(x)$ is computed through (5.4) using $u(x)$ and $a(x)$. In figure 7, the exact parameter q_{exact} (the red solid line) and the identified parameter $q_h(x)$ (the blue dashed line) are plotted when the observation noise levels are 0.01, 0.05 and 0.10, respectively. We note that the identifying source term $q(x)$ is several orders higher in magnitude ($O(10^2)$) than the state variable ($O(1)$): the source term varies between -300 to 400 on a unit interval. It is amazing that the MMC method performs quite well and gives a nice reconstruction in just four iterations.

Example 6. We take the diffusivity $a(x)$ and state variable $u(x)$ in (5.4) and the observed data $z^\delta(x)$ as follows,

$$a(x) = 7 + 2x^2 + 2\sin(2\pi x), \quad u(x) = \sin(2\pi x), \quad z^\delta(x) = (1 + \delta R(x))\sin(2\pi x),$$

and the source term $q(x)$ is computed through (5.4) using $u(x)$ and $a(x)$. In figure 8, the exact parameter q_{exact} (the red solid line) and the identified parameter $q_h(x)$ (the blue dashed line) are plotted when the error levels are 0.01, 0.05 and 0.10, respectively. We observe that the estimated parameter is still quite satisfactory for the current case with only the pointwise observation data.



(a)

(b)

(c)

Figure 7. Identification of source term in example 5: (a) $q_h^0 = 0, \delta = 0.01, \gamma = 5 \times 10^{-11}$, iter $= 4$, err $= 0.0051$; (b) $q_h^0 = 0, \delta = 0.05, \gamma = 5 \times 10^{-8}$, iter $= 3$, err $= 0.0074$; (c) $q_h^0 = 0, \delta = 0.10, \gamma = 7 \times 10^{-8}$, iter $= 3$, err $= 0.0089$.

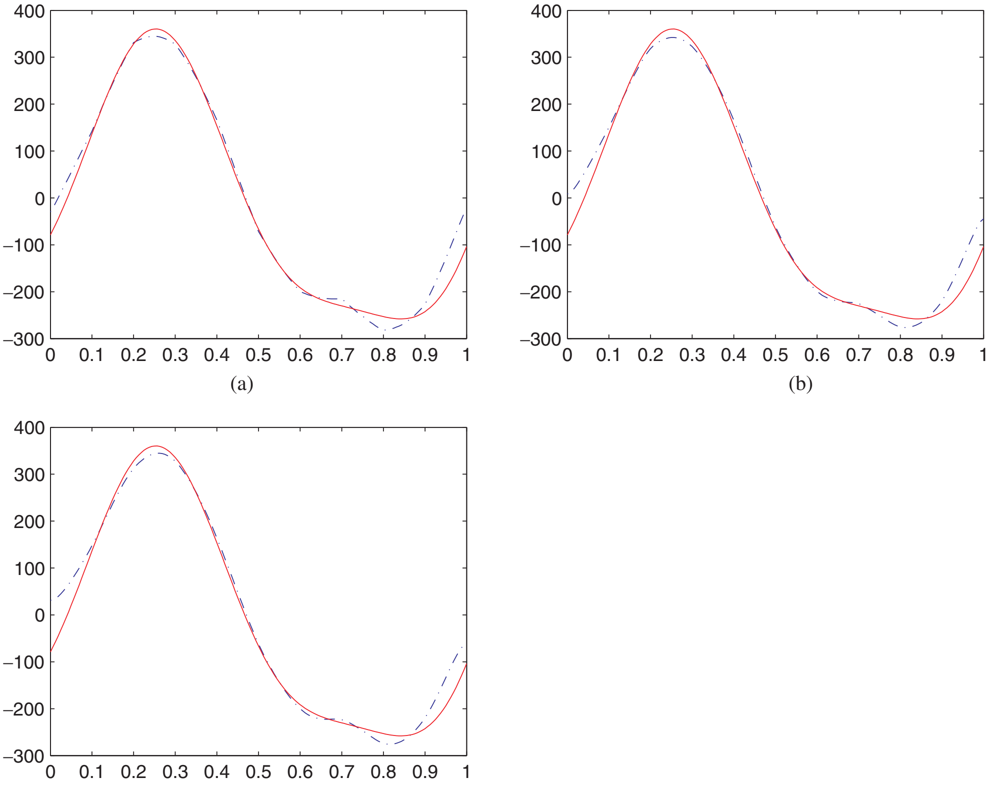
Example 7. We take the diffusivity $a(x, y)$, radiativity $q(x, y)$, state variable $u(x, y)$ in (5.5) and the observed data ∇z^δ as follows,

$$\begin{aligned} a(x, y) &= (1 + x^2 + yx)/1000, \\ q(x, y) &= 4 + \cos(\pi xy), \\ u(x, y) &= \sin(\pi x)\sin(\pi y), \\ \nabla z^\delta(x, y) &= (1 + \delta R(x, y))\nabla u(x, y), \end{aligned}$$

and the function $f(x, y)$ is computed through (5.5) using $u(x, y), a(x, y)$ and $q(x, y)$. Figure 9 presents the exact parameter q_{exact} and the identified parameter $q_h(x)$ when the noise levels are 0.01, 0.05 and 0.10, respectively. We observe that q is well identified, and the profile of the radiativity can be tracked by MMC in an efficient and stable way.

Example 8. We take the diffusivity $a(x, y)$, radiativity $q(x, y)$, state variable $u(x, y)$ in (5.5) and the observed data $z^\delta(x, y)$ as follows,

$$\begin{aligned} a(x, y) &= (1 + x^2 + yx)/1000, \\ q(x, y) &= 4 + \cos(\pi xy), \end{aligned}$$



(a)

(b)

(c)

Figure 8. Identification of source term in example 6: (a) $q_h^0 = 0, \delta = 0.01, \gamma = 1 \times 10^{-12}$, iter = 13, err = 0.01; (b) $q_h^0 = 0, \delta = 0.05, \gamma = 5 \times 10^{-11}$, iter = 17, err = 0.012; (c) $q_h^0 = 0, \delta = 0.10, \gamma = 1 \times 10^{-10}$, iter = 18, err = 0.016.

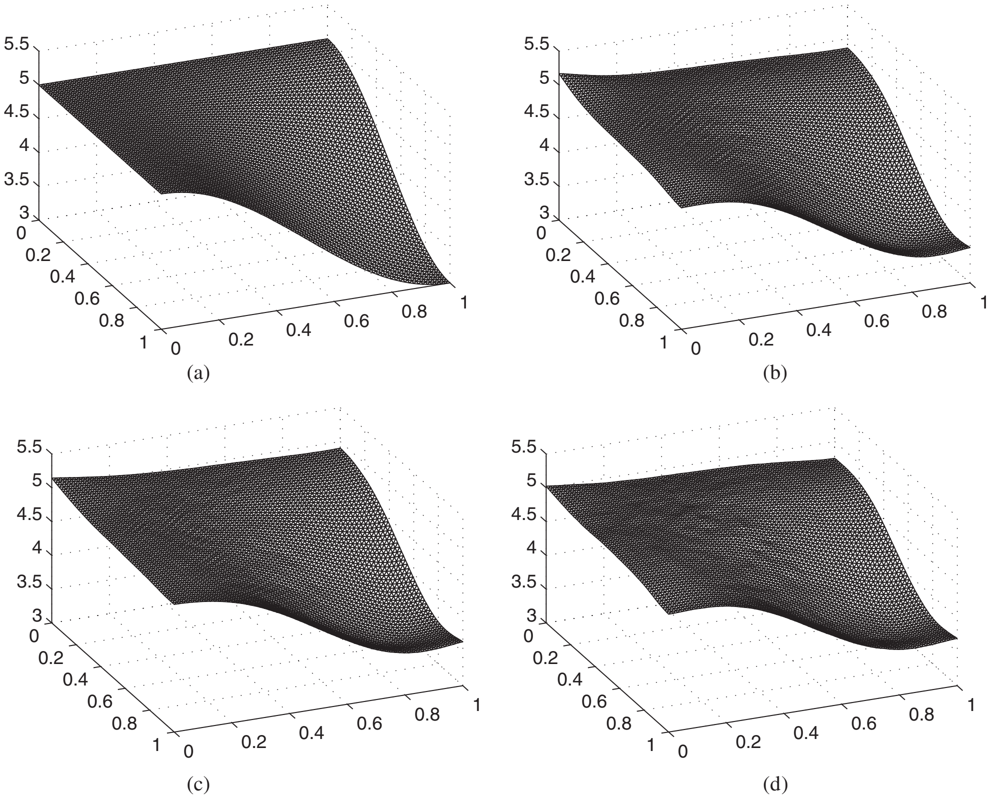
$$u(x, y) = \sin(\pi x)\sin(\pi y),$$
$$z^\delta(x, y) = (1 + \delta R(x, y))u(x, y),$$

and the function $f(x, y)$ is computed through (5.5) using $u(x, y), a(x, y)$ and $q(x, y)$. Figure 10 presents the exact parameter q_{exact} and the identified parameter $q_h(x)$ when the error levels are 0.01, 0.05 and 0.10, respectively. The identification process for the current case with L^2 observation data is much more ill-posed than the case with gradient data. But the profile of the radiativity can still be approximated well as in the previous example.

5.3. Some numerical observations

The numerical examples shown in section 5 have indicated that the proposed MMC algorithm is applicable to quite a number of parameter identification problems and the algorithm is robust with respect to the noise and initial guesses. In this section we shall report some further observations about the efficiency and behaviour of the MMC method.

First we plot the histories of the L^2-norm errors $\|q_k - q_{\text{exact}}\|$ between the current iterate of the MMC method and the exact solution, as well as the errors $\|q_k - q_{k-1}\|$ between two



(a)

(b)

(c)

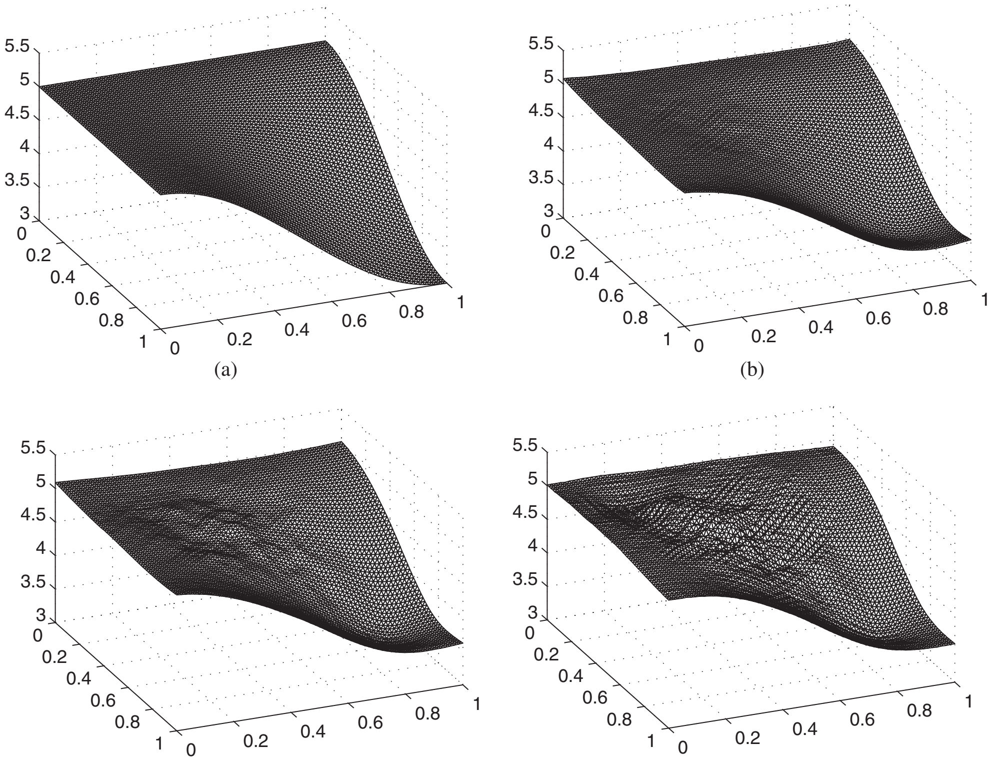
(d)

Figure 9. Identification of radiativity in example 7: (a) exact solution; (b) $q_h^0 = 10, \delta = 0.01, \gamma = 2 \times 10^{-6}$, iter = 6, err = 0.022; (c) $q_h^0 = 10, \delta = 0.05, \gamma = 3 \times 10^{-6}$, iter = 6, err = 0.031; (d) $q_h^0 = 10, \delta = 0.10, \gamma = 4 \times 10^{-6}$, iter = 6, err = 0.035.

consecutive iterates (with $q_{-1} = 0$) for examples 1 and 4. Similar phenomena are observed for other examples. From the semilog plot in figure 11, one can clearly see that the MMC converges fast and steadily to its best approximate solution, that is, the errors drop linearly on average till it reaches some critical point; then no further improvement on the numerical reconstruction can be achieved due to the discretization error and the noise in the observation data, although the MMC does not yet converge to its final solution.

We remark that the lower or upper bounds in the admissible sets, e.g., α_1 or α_2 in (4.6) and (4.7), should not be too close to the true solution, otherwise it may cause the algorithm to halt at the local extremes at the boundary of an admissible set. When this happens, one should relax the admissible sets with larger upper bounds or smaller lower bounds.

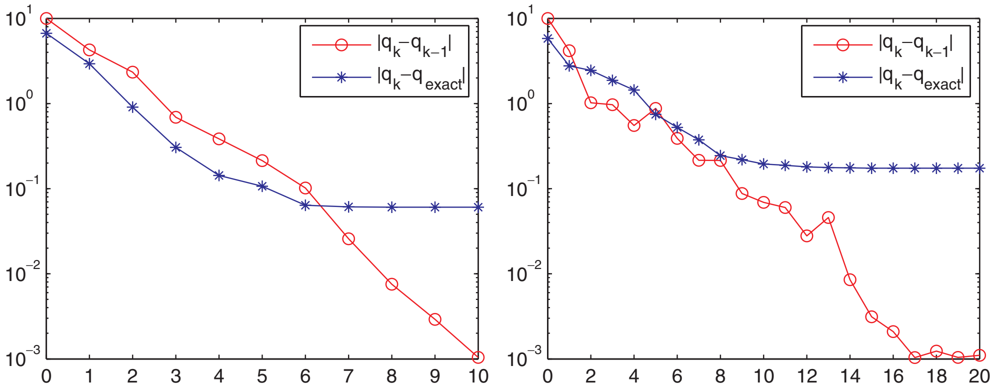
Next, we compare the CPU time (in seconds) of the MMC algorithm with the standard Armijo algorithm (cf [17]) for all the examples in section 5.2, see table 2. Armijo algorithms are stopped when the same L^2-norm errors are reached as indicated in the figures of section 5.2. As one can see from the statistics in table 2, significant speedups are achieved using the MMC algorithm; example 4 of identifying the diffusivity with L^2-data is the most difficult and time-consuming one; the time-stepping process of example 3 makes it very time-consuming even though the problem is one dimension in space. When the noise level is small



(a) (b)

(c) (d)

Figure 10. Identification of radiativity in example 8: (a) exact solution; (b) $q_h^0 = 10$, $\delta = 0.01$, $\gamma = 5 \times 10^{-6}$, iter = 10, err = 0.021; (c) $q_h^0 = 10$, $\delta = 0.05$, $\gamma = 6 \times 10^{-6}$, iter = 8, err = 0.025; (d) $q_h^0 = 10$, $\delta = 0.10$, $\gamma = 9 \times 10^{-6}$, iter = 6, err = 0.032.



(a) (b)

Figure 11. L^2-norm errors $\|q_k - q_{\text{exact}}\|$ (blue star line) and $\|q_k - q_{k-1}\|$ (red circle line) versus the number of iterations: (a) example 1 with $\delta = 0.05$; (b) example 4 with $\delta = 0.10$.

($\delta = 0.01$), we can achieve a better approximation with extra work, which explains why the CPU times are relatively smaller with large noise than with small noise.

Table 2. CPU time (in seconds) comparison between the MMC and Armijo algorithms.

Example no	1	2	3	4	5	6	7	8
				$\delta = 0.01$				
MMC	28.97	75.34	19.55	65.26	0.58	2.16	21.49	40.92
Armijo	423.04	2034.71	1407.82	1691.20	17.39	64.10	398.19	516.25
				$\delta = 0.05$				
MMC	26.72	72.65	15.81	59.40	0.55	2.47	20.85	33.97
Armijo	385.31	1942.10	1052.19	1546.16	17.12	72.38	365.30	485.62
				$\delta = 0.10$				
MMC	25.88	59.80	15.04	54.58	0.52	2.59	20.12	28.90
Armijo	374.69	1750.63	1034.38	1436.62	16.48	77.59	358.02	395.40

6. Concluding remarks

Output least-squares formulation with Tikhonov regularization is one of the most frequently used and reliable methods for parameter identifications in PDEs. This work proposes a multilevel model correction method, which aims at solving the general nonlinear output least-squares optimization system in an efficient and robust manner. Different from the usual numerical identification methods, the MMC method does not carry out the identification process on one mesh; instead it makes full use of a sequence of nested coarser meshes. It updates the output least-squares optimization functional and then solves the resulting optimization system at each level of mesh, recursively from fine to coarse meshes. This leads to a fast convergence of the reconstruction algorithm and satisfactory numerical identifications, even with high level of random noise in the observation data. The MMC method converges very fast, usually in a few iterations as illustrated in the numerical examples, to the best profile of the identifying parameters defined by the output least-squares formulation with Tikhonov regularization. We observe that the method is insensitive to the initial guesses, and it still converges very stably, even with bad initial guesses. Moreover, the MMC method is formulated in a very general and systematic manner so that it can be easily applied to many other inverse problems.

Acknowledgments

The authors would like to thank Professor Kazufumi Ito for the discussion of the convergence result during his visit at CUHK. We are also grateful to the anonymous referees for their very helpful comments. The work of JZ was substantially supported by Hong Kong RGC grants (Project 404105 and Project 404606) and was partially supported by Chang Jiang (Cheung Kong) Scholars Program through Wuhan University.

References

[1] Ascher U M and Haber E 2003 A multigrid method for distributed parameter estimation problems *Electr. Trans. Numer. Anal.* **15** 1–17

[2] Ascher U and Haber E 2001 Grid refinement and scaling for distributed parameter estimation problems *Inverse Problems* **17** 571–90

[3] Banks H T and Kunisch K 1989 *Estimation Techniques for Distributed Parameter Systems* (Boston, MA: Birkhäuser)

[4] Borcea L 2001 A nonlinear multigrid for imaging electrical conductivity and permittivity at low frequency *Inverse Problems* **17** 329–59

[5] Brandt A 1977 Multi-level adaptive solutions to boundary-value problems *Math. Comput.* **31** 333–90

[6] Chen Z M and Zou J 1999 An augmented Lagrangian method for identifying discontinuous parameters in elliptic systems *SIAM J. Control Optim.* **37** 892–910

[7] Ciarlet P G 1978 *The Finite Element Method for Elliptic Problems (1st edn) (Studies in Mathematics and its Applications)* (Amsterdam: North-Holland)

[8] Engl H W, Hanke M and Neubauer A 1996 *Regularization of Inverse Problems* (Dordrecht: Kluwer)

[9] Guenther R B, Hudspeth R, McDougal W and Gerlach J 1985 Remarks on parameter identification: I *Numer. Math.* **47** 355–61

[10] Haber E 2001 Quasi-Newton methods for large-scale electromagnetic inverse problems *Inverse Problems* **21** 305–23

[11] Hackbusch W 1985 *Multigrid Methods and Applications* (Berlin: Springer)

[12] Isakov V 1998 *Inverse Problems for Partial Differential Equations* (New York: Springer)

[13] Ito K and Kunisch K 1990 The augmented Lagrangian method for parameter estimation in elliptic systems *SIAM J. Control Optim.* **28** 113–36

[14] Kaltenbacher B 1997 Some Newton type methods for the regularization of nonlinear ill-posed problems *Inverse Problems* **13** 729–53

[15] Kaltenbacher B 2001 On the regularizing properties of a full multigrid method for ill-posed problems *Inverse Problems* **17** 767–88

[16] Kaltenbacher B and Schicho J 2002 A multi-grid method with *a priori* and *a posteriori* level choice for the regularization of nonlinear ill-posed problems *Numer. Math.* **93** 77–107

[17] Keung Y L and Zou J 1998 Numerical identifications of parameters in parabolic systems *Inverse Problems* **14** 83–100

[18] Keung Y L and Zou J 2000 An efficient linear solver for nonlinear parameter identification problems *SIAM J. Sci. Comput.* **22** 1511–26

[19] King J T 1992 Multilevel algorithms for ill-posed problems *Numer. Math.* **61** 311–34

[20] Kunisch K and Ring W 1993 Regularization of nonlinear ill-posed problems with closed operators *Numer. Funct. Anal. Optim.* **14** 389–404

[21] Lewis R M and Nash S G 2005 Model problems for the multigrid optimization of systems governed by differential equations *SIAM J. Sci. Comput.* **26** 1811–37

[22] Vogel C R 1999 Sparse matrix computations arising in distributed parameter identification *SIAM J. Matrix Anal. Appl.* **20** 1027–37

[23] Vogel C R 2000 A limited memory BFGS method for an inverse problem in atmospheric imaging *Methods and Applications of Inversion (Lecture Notes in Earth Sciences* vol 92*)* ed P C Hansen, B Jacobsen and K Mosegaard (Berlin: Springer) pp 292–304

[24] Xie J L and Zou J 2005 Numerical reconstruction of heat fluxes *SIAM J. Numer. Anal.* **43** 1504–35

[25] Yeh W W-G 1986 Review of parameter identification procedures in groundwater hydrology: the inverse problem *Water Resour. Res.* **22** 95–108

[26] Yamamoto M and Zou J 2001 Simultaneous reconstruction of the initial temperature and heat radiative coefficient *Inverse Problems* **17** 1181–202