Inverse Problems and Imaging
Volume 9, No. 1, 2015, 163–188
doi:10.3934/ipi.2015.9.163

OVERLAPPING DOMAIN DECOMPOSITION METHODS FOR LINEAR INVERSE PROBLEMS

Daijun Jiang

School of Mathematics and Statistics, Central China Normal University
Wuhan 430079, China

Hui Feng

School of Mathematics and Statistics, Wuhan University
Wuhan 430072, China

Jun Zou

Department of Mathematics, The Chinese University of Hong Kong
Shatin, Hong Kong, China

(Communicated by Xuecheng Tai)

Abstract. We shall derive and propose several efficient overlapping domain decomposition methods for solving some typical linear inverse problems, including the identification of the flux, the source strength and the initial temperature in second order elliptic and parabolic systems. The methods are iterative, and computationally very efficient: only local forward and adjoint problems need to be solved in each subdomain, and the local minimizations have explicit solutions. Numerical experiments are provided to demonstrate the robustness and efficiency of the methods: the algorithms converge globally, even with rather poor initial guesses; and their convergences do not deteriorate or deteriorate only slightly when the meshes are refined.

1. **Introduction.** Domain decomposition methods (DDMs) have been developed and proved to be one of the most successful methodologies in the construction of efficient numerical solvers for solving many boundary value and initial-boundary value problems, the so-called direct problems; see [14] [17] [18] and the references therein. DDMs usually possess two important features for solving a wide class of large-scale direct problems: first, they are natural parallel solvers and can be easily implemented in parallel computers; second, their convergence may be made nearly optimal in the sense that the resulting convergence rate is nearly independent of the mesh size.

However, no much progress has been made in the construction of efficient DDMs for solving mathematically ill-posed inverse problems, although the inverse problems are usually much more challenging and time consuming than their corresponding

2010 *Mathematics Subject Classification.* Primary: 31A25, 65M55; Secondary: 90C25.
Key words and phrases. Inverse problems, parameter identification, domain decomposition, explicit subdomain solver.
The first author was supported by self-determined research funds of CCNU from the colleges' basic research and operation of MOE (No. CCNU14A05039), National Natural Science Foundation of China (Nos. 11326233 and 11401241) and China Postdoctoral Science Foundation (Grant no. 2012M521444). The second author was supported by National Natural Science Foundation of China (No. 91130022, No. 10971159 and No. 11161130003) and NCET of China. The third author was substantially supported by Hong Kong RGC grants (Projects 404611 and 405513).

direct problems. In [5] [13], DDMs were applied indirectly for an elliptic identification problem, where classical iterative optimization algorithms were first adopted for solving the stabilized minimization system of the identification problem, then the existing DDMs were introduced for solving the direct problems and their adjoint systems involved at each iteration. This approach shares the standard convergence behaviours of the traditional iterative methods, that is, with the refinement of the mesh, the discretised minimization system becomes more ill-conditioned, and the number of iterations required for a given accuracy increases rapidly and can not be reduced no matter how the direct problem and its adjoint system involved are solved. Newton's method was used in [3] for solving the optimality system of the stabilized minimization of an elliptic identification problem, then an additive Schwarz type preconditioned algorithm was applied to solve the linear system involved at each Newton's iteration. As Newton's method requires the evaluations of the Hessian of the corresponding objective functional, the approach of [3] is applicable only to a very special formulation of the parameter identification problem, and as usual, the initial guesses are difficult to achieve for Newton's method. There are also efforts by DDMs for solving nonlinear optimizations arising, e.g., from optimal control problems with PDEs [6] [9] [10]. But these problems are quite different in nature from inverse problems, e.g., optimal control problems are usually well-posed and the (target) data is specified, mostly without any noise. In this work we shall develop some DDMs for directly solving the stabilized minimization systems of some typical linear inverse problems so that their convergences do not deteriorate or deteriorate only mildly as the entire degrees of freedom of the optimization system grow, and only subdomains are solved at each iteration and all subdomains are solved in parallel. Next, we shall briefly address some major difficulties in the construction of DDMs for inverse problems directly, then point out the new contributions of this work.

We shall use q and $u(q)$ to represent respectively the parameter function to be identified and the solution to the forward model system associated with the parameter q, then one may formulate a general inverse problem formally as the following forward operator equation

$$u(q) = z^\delta$$

where z^δ is the measured data of the exact solution u in some subregion inside the physical domain or on part of the boundary, or at the terminal time $t = T$ when the problem is time-dependent. And the parameter δ is used here to emphasize the existence of the noise in the measured data.

Inverse problems are usually ill-posed as at least one of the following three conditions is violated: the existence, uniqueness and stability of solutions [1][2][8]. Of the three conditions stability is the most frequently encountered difficulty in numerical solutions of inverse problems. One of the most stable and effective approaches to solve general ill-posed inverse problems is to transform them into stabilized output least-squares minimizations with some appropriately selected Tikhonov regularizations, namely to minimize the following type of functionals over some constrained set K:

$$J(q) = \|u(q) - z^\delta\|_V^2 + \beta N(q) \tag{1.1}$$

where V is a Hilbert or Banach space over the measurement subregion and is determined based on the type of measurement data available, $N(q)$ is the regularization

term and β is a regularization parameter to balance between the data fitting and regularization.

One of the major difficulties in the construction of DDMs for solving a nonlinear minimization problem associated with $J(q)$ lies in the global dependence of the forward operator $u(q)$ on the parameter q: a change of q in a small subregion of the global domain Ω causes the change of u in the entire Ω. This is generally true no matter if $u(q)$ is linear or nonlinear. Due to this global dependence, a direct application of the DDM principle to solve the nonlinear minimization problem of $J(q)$ may not work. To illustrate this point more clearly, we consider a decomposition of the global minimization of $J(q)$ over the entire domain Ω into a set of subproblems that involve only all sub-minimizations of functionals $J_i(q_i + \tilde{q})$ on the subdomains Ω_i, where q_i has support only in Ω_i, and $\tilde{q}$ is the known contribution from other subdomains, then $J_i(q_i + \tilde{q})$ should be of the form

$$J_i(q_i + \tilde{q}) = \|u(q_i + \tilde{q}) - z^\delta\|^2_{V_i} + \beta N(q_i + \tilde{q}). \tag{1.2}$$

Clearly the sub-minimization of functional J_i in (1.2) involves the solution $u(q_i+\tilde{q})$, which still needs to solve the forward problem in the global domain Ω even when operator $u(q)$ is linear and only the local quantity q_i needs to be updated. Hence the direct application of the DDMs does not really reduce the global computations to the ones in the local subdomains.

In this study, we will derive and propose several efficient overlapping DDMs for solving some typical linear inverse problems, including the identification of the source strength, the initial temperature inside a physical domain, and the fluxes on (inaccessible) part of the boundary of a physical domain in second order elliptic and parabolic systems. These inverse problems are all ill-posed, especially unstable with respect to the change of the noise in the data [2]. The new algorithms will be constructed in a way that meets the true spirits of DDMs, namely only sub-minimization problems are solved at each iteration on the subdomains of the original global domain, and the number of the iterations required for a specified accuracy grows nearly independent of (or very slowly on) the refinement of finite element meshes.

The rest of the paper is arranged as follows. In Section 2, we propose the Tikhonov regularization for identifying the source strength. In Section 2.1, the overlapping domain decomposition methods are first introduced and local minimizations are studied, then the algorithms are further improved. In Sections 3 and 4, we derive DDMs for the reconstruction of the fluxes on part of the boundary and the initial temperature inside a physical domain respectively. In Section 5, numerical experiments are presented for the identification of source strength, fluxes and initial temperature to illustrate the efficiency and robustness of the proposed algorithms. Some concluding remarks are given in Section 6.

Throughout the paper, C is often used for a generic constant. We shall use the symbol $\langle\cdot,\cdot\rangle$ for the general inner product, and write the norms of the spaces $H^m(\Omega)$, $L^2(\Omega)$, $H^{\frac{1}{2}}(\Gamma)$ and $L^2(\Gamma)$ (for some $\Gamma \subset \partial\Omega$) respectively as $\|\cdot\|_{m,\Omega}$, $\|\cdot\|_\Omega$, $\|\cdot\|_{1/2,\Gamma}$ and $\|\cdot\|_\Gamma$.

2. Domain decomposition algorithms for the reconstruction of source strengths.

The major task of this work is to propose some new overlapping DDMs for solving three typical linear inverse problems, including the identification of the source strength, the flux and the initial temperature. For ease of exposition, we shall take the inverse problem of identifying the source strength in a diffusion system as

an example to derive and discuss the new DDMs in more detail in this section, and address the other two inverse problems in sections 3 and 4. Let Ω be an open bounded and connected domain in $\mathbb{R}^d$ ($d \geq 1$), with a boundary $\partial\Omega$. Then we consider the following diffusion system

$$\left\{ \begin{array}{rcll} -\nabla \cdot (a(\mathbf{x})\nabla u) + c(\mathbf{x})u &=& f(\mathbf{x}) & \text{in} \quad \Omega, \\ u(\mathbf{x}) &=& g(\mathbf{x}) & \text{on} \quad \partial\Omega \end{array} \right. \tag{2.1}$$

where $a(\mathbf{x})$, $c(\mathbf{x})$ and $g(\mathbf{x})$ are all given functions, and $a(\mathbf{x}) \geq a_1 > 0$, $c(\mathbf{x}) \geq c_1 > 0$ in Ω. Suppose that the source strength $f(\mathbf{x})$ of the model system is unknown in Ω. Our inverse problem is to recover the source strength distribution $f(\mathbf{x})$ in Ω when the measurement data of u, denoted by z^δ, is available in Ω, or in a subregion $\tilde{\Omega}$ of Ω. For convenience, we shall write the solution of system (2.1) as $u(f)$ to emphasize its dependence on the source strength $f(\mathbf{x})$. This is a well-known ill-posed problem. As in (1.1), we formulate it in a mathematically stabilized minimization system of the form

$$\min_{f \in L^2(\Omega)} J(f) \quad = \quad \|u(f) - z^\delta\|_\Omega^2 + \beta \|f\|_\Omega^2 \,. \tag{2.2}$$

One can show that the minimizer of the system is stable in the sense that it depends continuously on the change of the noise in the data z^δ [11] [16].

Linearity of the forward solutions. The forward solution $u(f)$ of the system (2.1) is basically linear in terms of f. It is easy to check directly that

$$u(\lambda_1 f_1 + \lambda_2 f_2) = \lambda_1 u(f_1) + \lambda_2 u(f_2) \quad \forall\, f_1, f_2 \in L^2(\Omega) \text{ and } \lambda_1, \lambda_2 \in R$$

if and only if $g(\mathbf{x}) = 0$. This leads us to consider the solution U to the following system:

$$\left\{ \begin{array}{rcll} -\nabla \cdot (a(\mathbf{x})\nabla U) + c(\mathbf{x})U &=& f(\mathbf{x}) & \text{in} \quad \Omega, \\ U &=& 0 & \text{on} \quad \partial\Omega \,. \end{array} \right. \tag{2.3}$$

We can verify that $u(f_1) - u(f_2) = U(f_1 - f_2)$ for any $f_1, f_2 \in L^2(\Omega)$, or equivalently we have

$$u(f) = U(f) + u(0). \tag{2.4}$$

From now on we shall view the solution $U(f)$ to (2.3) as a mapping from $L^2(\Omega)$ to $L^2(\Omega)$.

Adjoint operator. It is easy to verify that operator $U(f)$ is self-adjoint. In fact, we have by integration by parts for any $\omega \in L^2(\Omega)$ that

$$\begin{aligned} \langle f, U(\omega) \rangle_{L^2(\Omega)} &= \langle -\nabla \cdot (a(\mathbf{x})\nabla U(f)) + c(\mathbf{x})U(f), U(\omega) \rangle_{L^2(\Omega)} \\ &= \langle U(f), -\nabla \cdot (a(\mathbf{x})\nabla U(\omega)) + c(\mathbf{x})U(\omega) \rangle_{L^2(\Omega)} \\ &= \langle U(f), \omega \rangle_{L^2(\Omega)}. \end{aligned} \tag{2.5}$$

2.1. **Overlapping DDMs with explicit local solvers.** Using the relation (2.4) we can rewrite the minimization (2.2) as

$$\min_{f \in L^2(\Omega)} J(f) = \|U(f) - z_0^\delta\|_\Omega^2 + \beta \|f\|_\Omega^2 \,, \tag{2.6}$$

with $z_0^\delta = z^\delta - u(0)$. As $U(f)$ is linear, $J(f)$ is convex with respect to f. And the minimizers of (2.6) exist and are unique.

Next, we shall derive some effective DDMs to solve the optimization system (2.6). We shall not intend to solve this optimization system on the global domain Ω, as most existing numerical solvers do. Instead we plan to construct some DDMs so that

the nonlinear system (2.6) can be effectively solved on local subdomains. To do so, we divide the global domain Ω into a finite number of overlapping subdomains Ω_1, Ω_2, ... , Ω_l, where l is a positive integer. Though our new DDMs work for a general number of subdomains, we shall focus all our discussions only on 4 subdomains with a cross-point for ease of exposition; see Figure 2.1. It is well-known that the case of 4 subdomains with a cross-point is the most representative case of general multiple subdomains [14] [18].

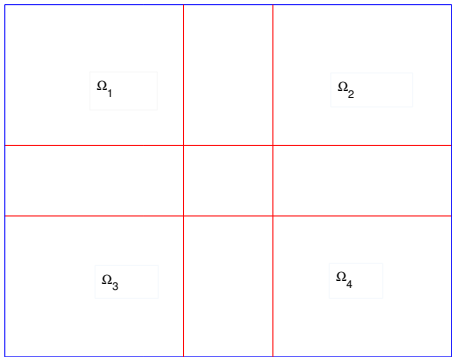

FIGURE 2.1. Domain Ω with its 4 overlapping subdomains $\Omega_1, \Omega_2, \Omega_3, \Omega_4$

Based on the partition of Ω into 4 overlapping subdomains, we shall often need a local subspace of $L^2(\Omega)$ on each subdomain Ω_i:

$$V_i = \Big\{ f \in L^2(\Omega);\ \text{supp}(f) \subset \Omega_i \Big\}, \quad i = 1,\ 2,\ 3,\ 4.$$

Next we start to derive some new DD algorithms for solving the optimization system (2.6). The algorithms are based on the local optimizations on the subspaces V_i associated with subdomain Ω_i. For some given $f_j \in V_j$ $(j = 1, 2, 3, 4)$, let us consider the following local minimization over Ω_i:

$$\min_{v_i \in V_i} J\Big(v_i + \sum_{j \neq i} f_j\Big). \tag{2.7}$$

Here and in the sequel, we often write $\sum_{j=1, j\neq i}^{4}$ as $\sum_{j\neq i}$ for simplicity. By the definition of J in (2.6) we know that each local update v_i in Ω_i still needs to compute the quantity $U(v_i + \sum_{j\neq i} f_j)$, which involves the solution of the forward system (2.3) in the entire domain Ω. To avoid this, we construct an auxiliary functional $\tilde{J}_i^s$ of J, called the surrogate functional in [7], by introducing an auxiliary variable a. For a given $a \in V_i$ and $f_j \in V_j$ $(j = 1, 2, 3, 4)$, we define

$$\tilde{J}_i^s(\sum_{j=1}^{4} f_j, a) = J(\sum_{j=1}^{4} f_j) + A\,\|f_i - a\|_\Omega^2 - \|U(f_i - a)\|_\Omega^2 \tag{2.8}$$

where A is a positive constant to be selected such that

$$A\|f_i - a\|_\Omega^2 - \|U(f_i - a)\|_\Omega^2 \geq (A - \|U\|^2)\|f_i - a\|_\Omega^2 \geq 0\,. \tag{2.9}$$

This implies for any $f = \sum_{j=1}^4 f_j$ that $\tilde{J}_i^s(f,a) = J(f)$ when $a = f_i$, and

$$J(f) = \tilde{J}_i^s(f,f_i) \le \tilde{J}_i^s(f,a) = J(f) + A\|f_i - a\|_\Omega^2 - \|U(f_i - a)\|_\Omega^2 \quad \forall\, a \in V_i\,. \tag{2.10}$$

So $\tilde{J}_i^s(f,a)$ can be viewed as a small perturbation of $J(f)$ when a is close to f_i.

In order to justify the surrogate functional, we can naturally extend the arguments in [7] to prove the following convergence of the iteration suggested by the surrogate functional.

Lemma 2.1. *Suppose A is a constant such that $A\|f\|_\Omega^2 \ge \|U(f)\|_\Omega^2$ for $f \in V_i$, then the following sequence produced by the surrogate functional (2.8) for any $f_i^0 \in V_i$ and $n = 1, 2, \cdots$,*

$$f_i^n \;=\; \operatorname{argmin}_{f_i \in V_i} J\Big(f_i + \sum_{j\neq i} f_j\Big) + A\|f_i - f_i^{n-1}\|_\Omega^2 - \|U(f_i - f_i^{n-1})\|_\Omega^2$$

converges strongly to the minimizer of system (2.7).

Now we shall convert (2.8) into a more explicit representation. Using (2.6), (2.8) and the adjoint relation (2.5) we can rewrite $\tilde{J}_i^s$ as follows:

$$\begin{aligned}
&\tilde{J}_i^s(\sum_{j=1}^4 f_j, a)\\
=\;& \|U(f_i)\|_\Omega^2 - 2\langle f_i,\, U(z_0^\delta - U(\sum_{j\neq i} f_j)\rangle_\Omega + \|z_0^\delta - U(\sum_{j\neq i} f_j)\|_\Omega^2 + \beta\|\sum_{j=1}^4 f_j\|_\Omega^2\\
&+A\langle f_i,\, f_i - 2a\rangle_\Omega + A\|a\|_\Omega^2 - \|U(f_i)\|_\Omega^2 + 2\langle f_i,\, U(U(a))\rangle_\Omega - \|U(a)\|_\Omega^2\\
=\;& A\Big\langle f_i,\, f_i - 2\Big\{a + \frac{1}{A}U\Big(z_0^\delta - U(\sum_{j\neq i} f_j) - U(a)\Big)\Big\}\Big\rangle_{\Omega_i}\\
&+\beta\|\sum_{j=1}^4 f_j\|_\Omega^2 + \|z_0^\delta - U(\sum_{j\neq i} f_j)\|_\Omega^2 + A\|a\|_\Omega^2 - \|U(a)\|_\Omega^2\\
=\;& A\|f_i - \Big\{a + \frac{1}{A}U\Big(z_0^\delta - U(\sum_{j\neq i} f_j + a)\Big)\Big\}\|_{\Omega_i}^2 + \beta\|\sum_{j=1}^4 f_j\|_\Omega^2\\
&+\Big\{\|z_0^\delta - U(\sum_{j\neq i} f_j)\|_\Omega^2 + A\|a\|_\Omega^2 - \|U(a)\|_\Omega^2\\
&-A\|a + \frac{1}{A}U\Big(z_0^\delta - U(\sum_{j\neq i} f_j + a)\Big)\|_{\Omega_i}^2\Big\}\,.
\end{aligned} \tag{2.11}$$

We can see that the last term in (2.11) does not depend on f_i, so it will not affect the local minimization over Ω_i if we drop it in the functional $\tilde{J}_i^s$. This leads us to consider the following functional for a given $a \in V_i$:

$$\min_{f_i \in V_i} \tilde{J}_i^s(f_i + \sum_{j\neq i} f_j, a) = \min_{f_i \in V_i} A\|f_i - \tilde{z}_i\|_{\Omega_i}^2 + \beta\|\sum_{j=1}^4 f_j\|_\Omega^2 \tag{2.12}$$

where $\tilde{z}_i$ is given by

$$\tilde{z}_i = a + \frac{1}{A}U(z_0^\delta - U(\sum_{j\neq i} f_j + a)). \tag{2.13}$$

Noting that (2.12) is a simple quadratic minimization, we can find its exact minimizer f_i^*:

$$f_i^* = \frac{1}{A+\beta}\Big(A\tilde{z}_i - \beta\sum_{j\neq i} f_j\Big)|_{\Omega_i}. \tag{2.14}$$

Clearly, the new functional $\tilde{J}_i^s$ in (2.12) has an obvious advantage over the functional J in (2.6) or (2.7): it is completely local, and the minimization can be solved explicitly within the subdomain Ω_i. However, for the solution of the local minimization (2.12) we need the data $\tilde{z}_i$ from (2.13), which involves the evaluations of $U(\sum_{j\neq i} f_j + a)$ and $U(z_0^\delta - U(\sum_{j\neq i} f_j + a))$. Unfortunately, these two evaluations are both global, and require the solutions of the forward system (2.3) in the entire domain Ω. This is surely not expected in an efficient DD algorithm.

Next, we shall propose some techniques to get rid of the aforementioned two global evaluations so that the resulting DD algorithm involves only local minimizations over the local subdomains. For convenience, we write the boundary of Ω_i inside Ω by $\tilde{\Gamma}_i$, i.e., $\tilde{\Gamma}_i = \partial\Omega_i \cap \Omega$ for $i = 1, 2, 3, 4$. Then we introduce a local forward operator $U_i(f, p)$ associated with the forward problem (2.3):

$$\left\{\begin{array}{rcccc} -\nabla\cdot(a(\mathbf{x})\nabla U_i(f,\,p)) + c(\mathbf{x})U_i(f,\,p) &=& f & \text{in} & \Omega_i, \\ U_i(f,p) &=& 0 & \text{on} & \partial\Omega\cap\partial\Omega_i \\ U_i(f,p) &=& p & \text{on} & \tilde{\Gamma}_i. \end{array}\right. \tag{2.15}$$

Clearly we can split $U_i(f, p)$ as $U_i(f, p) = U_i(f, 0) + U_i(0, p)$, and $U_i(f, 0)$ is self-adjoint, i.e.,

$$\langle U_i(f,0),\,\omega\rangle_{\Omega_i} \;=\; \langle f,\,U_i(\omega,0)\rangle_{\Omega_i} \quad \forall\,\omega\in L^2(\Omega_i). \tag{2.16}$$

Using the local operators $U_i(f, p)$ in (2.15), we introduce the following local functional for $f_j \in V_j$, $j = 1, 2, 3, 4$:

$$J_i(\sum_{j=1}^4 f_j, p) \;=\; \|U_i(\sum_{j=1}^4 f_j, p) - z_0^\delta\|_{\Omega_i}^2 + \beta\|\sum_{j=1}^4 f_j\|_{\Omega_i}^2,$$

and its surrogate functional J_i^s for any given $a \in V_i$:

$$J_i^s(\sum_{j=1}^4 f_j, p, a) = J_i(\sum_{j=1}^4 f_j, p) + A\|f_i - a\|_{\Omega_i}^2 - \|U_i(f_i - a, 0)\|_{\Omega_i}^2.$$

Using the important fact that $U_i(\sum_{j=1}^4 f_j, p) = U_i(\sum_{j\neq i} f_j, p) + U_i(f_i, 0)$ and the adjoint relation (2.16), we can write

$$\begin{aligned} & J_i^s(\sum_{j=1}^4 f_j, p, a) \\ = \;& \|U_i(f_i,0)\|_{\Omega_i}^2 - 2\langle f_i,\,U_i\Big(z_0^\delta - U_i(\sum_{j\neq i} f_j, p), 0\Big)\rangle_{\Omega_i} \\ & + \|z_0^\delta - U_i(\sum_{j\neq i} f_j, p)\|_{\Omega_i}^2 + \beta\|\sum_{j=1}^4 f_j\|_{\Omega_i}^2 + A\langle f_i,\,f_i - 2a\rangle_{\Omega_i} + A\|a\|_{\Omega_i}^2 \\ & - \|U_i(f_i,0)\|_{\Omega_i}^2 + 2\langle f_i,\,U_i(U_i(a,0),0)\rangle_{\Omega_i} - \|U_i(a,0)\|_{\Omega_i}^2 \end{aligned}$$

$$
\begin{aligned}
&= A\langle f_i,\, f_i - 2\Big\{a + \frac{1}{A}U_i\Big(z_0^\delta - U_i(\sum_{j\neq i} f_j + a, p), 0\Big)\Big\}\rangle_{\Omega_i} + \beta\|\sum_{j=1}^4 f_j\|^2_{\Omega_i} \\
&\quad + \|z_0^\delta - U_i(\sum_{j\neq i} f_j, p)\|^2_{\Omega_i} + A\|a\|^2_{\Omega_i} - \|U_i(a,0)\|^2_{\Omega_i} \\
&= A\|f_i - \Big\{a + \frac{1}{A}U_i\Big(z_0^\delta - U_i(\sum_{j\neq i} f_j + a, p), 0\Big)\Big\}\|^2_{\Omega_i} + \beta\|\sum_{j=1}^4 f_j\|^2_{\Omega_i} \\
&\quad + \Big\{\|z_0^\delta - U_i(\sum_{j\neq i} f_j, p)\|^2_{\Omega_i} + A\|a\|^2_{\Omega_i} - \|U_i(a,0)\|^2_{\Omega_i} \\
&\quad - A\|a + \frac{1}{A}U_i\Big(z_0^\delta - U_i(\sum_{j\neq i} f_j + a, p), 0\Big)\|^2_{\Omega_i}\Big\}. \qquad (2.17)
\end{aligned}
$$

We can easily see that the last term above does not depend on f_i, so it will not affect the local minimization over Ω_i if we drop it in the functional J_i^s. This leads us to consider the following functional for a given $a \in V_i$:

$$
\min_{f_i\in V_i} J_i^s(\sum_{j=1}^4 f_j, p, a) = \min_{f_i\in V_i} A\|f_i - z_i\|^2_{\Omega_i} + \beta\|\sum_{j=1}^4 f_j\|^2_{\Omega_i}, \qquad (2.18)
$$

where $z_i = a + \frac{1}{A}U_i\Big(z_0^\delta - U_i(\sum_{j\neq i} f_j + a, p), 0\Big)$. (2.18) is a simple quadratic minimization, and we can find its exact minimizer f_i^*:

$$
f_i^* = \frac{1}{A+\beta}\Big\{A\,a + U_i\Big(z_0^\delta - U_i(\sum_{j\neq i} f_j + a, p), 0\Big) - \beta\sum_{j\neq i} f_j|_{\Omega_i}\Big\}. \qquad (2.19)
$$

We can see from this expression that as long as the inner boundary value p is available, the minimization (2.18) does not involve any global data and is completely local. Noting that $U(f)|_{\Omega_i} = U_i(f, U(f))$ and the definitions of $J_i(f,p)$ and $J_i^s(f,p,a)$, we can connect $J_i(f,p)$ and $J_i^s(f,p,a)$ with functional $J(f)$ (cf. (2.6)) restricted in Ω_i:

$$
\begin{aligned}
&\|U(\sum_{j=1}^4 f_j) - z_0^\delta\|^2_{\Omega_i} + \beta\|\sum_{j=1}^4 f_j\|^2_{\Omega_i} \\
&= \|U_i(\sum_{j=1}^4 f_j, U(\sum_{j=1}^4 f_j)) - z_0^\delta\|^2_{\Omega_i} + \beta\|\sum_{j=1}^4 f_j\|^2_{\Omega_i} \\
&= J_i(\sum_{j=1}^4 f_j, U(\sum_{j=1}^4 f_j)) = J_i^s(\sum_{j=1}^4 f_j, U(\sum_{j=1}^4 f_j), f_i). \qquad (2.20)
\end{aligned}
$$

So using (2.18), we are now ready to apply the multiplicative or additive Schwarz principle [14] [18] to establish two DD algorithms for solving the optimization system (2.6). For the description of the algorithms, we introduce an index function for any point $\mathbf{x} \in \Omega$:

(2.21)

$$
n(\mathbf{x}) = \Big\{i;\, \mathbf{x}\in\Omega_i,\ i\in\{1,2,3,4\}\Big\};\quad |n(\mathbf{x})| = \text{ number of elements in } n(\mathbf{x})\,.
$$

Algorithm 2.1 (Multiplicative Schwarz Algorithm (MSA)). *Choose a tolerance parameter $\epsilon_1 > 0$, an initial value $f^{(0)} = \sum_{i=1}^4 f_i^{(0)}$ with $f_i^{(0)} \in V_i$ $(i = 1,2,3,4)$, and solve (2.3) for $U(f^{(0)})$; set $p_i^{(0)} := U(f^{(0)})|_{\tilde{\Gamma}_i}$ and $n := 0$.*

1. *Compute $f_i^{(n+1)} \in V_i$ sequentially for $i = 1$ to 4 by*

$$f_i^{(n+1)} = \operatorname{argmin}_{v_i \in V_i} J_i^s(\sum_{j<i} f_j^{(n+1)} + v_i + \sum_{j>i} f_j^{(n)}, p_i^{(n)}, f_i^{(n)}); \tag{2.22}$$

update U_i in Ω_i:

$$U_i^{(n)} = U_i(\sum_{j\le i} f_j^{(n+1)} + \sum_{j>i} f_j^{(n)}, p_i^{(n)});$$

update the inner boundary values on $\tilde{\Gamma}_j$ for $j > i$ if $\tilde{\Gamma}_j \in \Omega_i$:

$$p_j^{(n)} = U_i^{(n)}|_{\tilde{\Gamma}_j}.$$

2. *Compute $f^{(n+1)} = \sum_{i=1}^4 f_i^{(n+1)}$.*
3. *If $\|f^{(n+1)} - f^{(n)}\|_\Omega \le \epsilon_1$, stop the iteration; otherwise update U_i in subdomain Ω_i $(i = 1,2,3,4)$:*

$$U_i^{(n+1)} = U_i(f^{(n+1)}, p_i^{(n)});$$

update the inner boundary values on $\tilde{\Gamma}_i$ $(i = 1,2,3,4)$:

$$p_i^{(n+1)}(\mathbf{x}) = \frac{1}{|n(\mathbf{x})|} \sum_{j \in n(\mathbf{x})} U_j^{(n+1)}(\mathbf{x}), \quad \forall \mathbf{x} \in \tilde{\Gamma}_i.$$

set $n := n + 1$, go to Step 1.

We can easily see that Algorithm 2.1 is sequential or multiplicative. The next algorithm proposes a parallel version of Algorithm 2.1. For this purpose, we introduce a bounded uniform partition of unity $\{\chi_i\}_{i=1}^4$ such that $\sum_{i=1}^4 \chi_i = 1$ and $\|\chi_i\|_\infty \le 1$ and $\operatorname{supp}(\chi_i) \subset \Omega_i$.

Algorithm 2.2 (Additive Schwarz Algorithm (ASA)). *Choose a tolerance parameter $\epsilon_1 > 0$, a relaxation parameter $\lambda \in (0,1)$, an initial value $f^{(0)} = \sum_{i=1}^4 f_i^{(0)}$ with $f_i^{(0)} \in V_i$ $(i = 1,2,3,4)$, and solve (2.3) for $U(f^{(0)})$; set $p_i^{(0)} := U(f^{(0)})|_{\tilde{\Gamma}_i}$ and $n := 0$.*

1. *Compute $f_i^{(n+1)} \in V_i$ in parallel for $i = 1,2,3,4$ by*

$$f_i^{(n+1)} = \operatorname{argmin}_{v_i \in V_i} J_i^s(\sum_{j\ne i} f_j^{(n)} + v_i, p_i^{(n)}, f_i^{(n)}). \tag{2.23}$$

2. *Compute $f^{(n+1)} = \lambda \sum_{i=1}^4 f_i^{(n+1)} + (1-\lambda) f^{(n)}$.*
3. *If $\|f^{(n+1)} - f^{(n)}\|_\Omega \le \epsilon_1$, stop the iteration; otherwise update U_i in subdomains Ω_i $(i = 1,2,3,4)$:*

$$U_i^{(n+1)} = U_i(f^{(n+1)}, p_i^{(n)});$$

update the inner boundary values on $\tilde{\Gamma}_i$ $(i = 1,2,3,4)$:

$$p_i^{(n+1)}(\mathbf{x}) = \frac{1}{|n(\mathbf{x})|} \sum_{j \in n(\mathbf{x})} U_j^{(n+1)}(\mathbf{x}) \quad \forall \mathbf{x} \in \tilde{\Gamma}_i.$$

set $f_i^{(n+1)} := \chi_i f^{(n+1)}$, and $n := n + 1$, go to Step 1.

Remark 2.1. The same as for (2.19), we have explicit expressions for the minimizers $f_i^{(n+1)}$ in (2.22) and (2.23). In our numerical implementations, we will simply take the partition of unity $\{\chi_i\}_{i=1}^4$ used in Algorithm 2.2 as follows:

$$\chi_i(\mathbf{x}) = 1/|n(\mathbf{x})| \quad \text{for} \quad \mathbf{x} \in \Omega_i\,; \quad \chi_i(\mathbf{x}) = 0 \quad \text{for} \quad \mathbf{x} \in \Omega \backslash \bar{\Omega}_i.$$

3. **Domain decomposition algorithms for flux reconstruction.** In this section, we propose a DD algorithm to solve the inverse problem of identifying fluxes on part of the boundary. Let $\Omega \subset R^d$ $(d \geq 1)$ be an open bounded and connected domain, with a boundary $\partial\Omega$, which is split into two parts, i.e., $\partial\Omega = \Gamma_0 \cup \Gamma_1$. Then we consider the following elliptic system

$$\left\{ \begin{array}{rcll} -\nabla \cdot (a(\mathbf{x})\nabla u) + c(\mathbf{x})u &=& f(\mathbf{x}) & \text{in} \quad \Omega\,, \\ a(\mathbf{x})\frac{\partial u}{\partial n} &=& g(\mathbf{x}) & \text{on} \quad \Gamma_0\,, \\ a(\mathbf{x})\frac{\partial u}{\partial n} &=& h(\mathbf{x}) & \text{on} \quad \Gamma_1\,, \end{array} \right. \tag{3.1}$$

where $a(\mathbf{x})$, $c(\mathbf{x})$, $f(\mathbf{x})$, $g(\mathbf{x})$ are all given functions, and $a(\mathbf{x}) \geq a_1 > 0$, $c(\mathbf{x}) \geq c_1 > 0$ in Ω. Suppose that the flux $h(\mathbf{x})$ of the model system is unknown on the inaccessible part Γ_1 of $\partial\Omega$, our inverse problem is to recover the flux distribution on Γ_1 when some measurement data u^δ of u is available on the accessible part Γ_0 of $\partial\Omega$. We shall write the solution of system (3.1) as $u(h)$ to emphasize its dependence on the flux $h(\mathbf{x})$.

As discussed in section 2, we formulate the ill-posed inverse problem of recovering the flux into a mathematically stabilized minimization system of the form

$$\min_{h \in L^2(\Gamma_1)} J(h) \quad = \quad \|u(h) - z^\delta\|_{\Gamma_0}^2 + \beta \|h\|_{\Gamma_1}^2\,. \tag{3.2}$$

This formulation is stable in the sense that the minimizer h to (3.2) depends continuously on the change of the noise in the data u^δ [16].

Similarly to the discussions in Section 2, we can write the solution $u(h)$ to (3.1) as

$$u(h) = U(h) + u(0)\,, \tag{3.3}$$

where $U(h)$ is the solution to the following system:

$$\left\{ \begin{array}{rcll} -\nabla \cdot (a(\mathbf{x})\nabla U) + c(\mathbf{x})U &=& 0 & \text{in} \quad \Omega, \\ a(\mathbf{x})\frac{\partial U}{\partial n} &=& 0 & \text{on} \quad \Gamma_0, \\ a(\mathbf{x})\frac{\partial U}{\partial n} &=& h & \text{on} \quad \Gamma_1\,. \end{array} \right. \tag{3.4}$$

Adjoint operator of $U(h)$. For any $\omega \in L^2(\Gamma_0)$, consider the solution $U^*(\omega) \in H^1(\Omega)$ to the following system:

$$\left\{ \begin{array}{rcll} -\nabla \cdot (a(\mathbf{x})\nabla U^*(\omega)) + c(\mathbf{x})U^*(\omega) &=& 0 & \text{in} \quad \Omega, \\ a(\mathbf{x})\frac{\partial U^*(\omega)}{\partial n} &=& \omega & \text{on} \quad \Gamma_0, \\ a(\mathbf{x})\frac{\partial U^*(\omega)}{\partial n} &=& 0 & \text{on} \quad \Gamma_1. \end{array} \right. \tag{3.5}$$

This mapping $U^* : L^2(\Gamma_0) \to L^2(\Gamma_1)$ is the adjoint operator of U, namely, it holds that

$$\langle U(h), \omega \rangle_{\Gamma_0} = \langle h, U^*(\omega) \rangle_{\Gamma_1} \quad \forall\, \omega \in L^2(\Gamma_0). \tag{3.6}$$

This relation follows directly from (3.4), (3.5) and an application of integration by parts:

$$\begin{aligned}
&\langle U(h),\, \omega\rangle_{\Gamma_0} = \langle U(h), a(\mathbf{x})\frac{\partial U^*(\omega)}{\partial n}\rangle_{\Gamma_0}\\
=\;& \langle U(h),\, a(\mathbf{x})\frac{\partial U^*(\omega)}{\partial n}\rangle_{\partial\Omega} + \int_\Omega U(h)(-\nabla\cdot(a(\mathbf{x})\nabla U^*(\omega)) + c(\mathbf{x})U^*(\omega))d\mathbf{x}\\
=\;& \int_\Omega (-\nabla\cdot(a(\mathbf{x})\nabla U(h)) + c(\mathbf{x})U(h))U^*(\omega)d\mathbf{x} + \langle a(\mathbf{x})\frac{\partial U(h)}{\partial n},\, U^*(\omega)\rangle_{\partial\Omega}\\
=\;& \langle a(\mathbf{x})\frac{\partial U(h)}{\partial n},\, U^*(\omega)\rangle_{\Gamma_1} = \langle h,\, U^*(\omega)\rangle_{\Gamma_1}.
\end{aligned}$$

3.1. **Overlapping DDMs with explicit local solvers.** In this subsection, we follow section 2.1 to derive some overlapping domain decomposition method for solving the minimization in (3.2). As in section 2.1, Ω is divided into the overlapping subdomains Ω_i $(i = 1, 2, 3, 4)$, accordingly the feasible constraint space $L^2(\Gamma_1)$ can be decomposed into the subspaces

$$V_i = \Big\{h \in L^2(\Gamma_1);\ \mathrm{supp}(h) \subset \partial\Omega_i \cap \Gamma_1\Big\}, \qquad i = 1,\, 2,\, 3,\, 4.$$

Next we introduce an auxiliary surrogate functional $\tilde{J}_i^s$ of $J(h)$ in (3.2) for any given $a \in V_i$ and $h_j \in V_j$ $(j = 1, 2, 3, 4)$:

$$\tilde{J}_i^s(\sum_{j=1}^4 h_j, a) = J(\sum_{j=1}^4 h_j) + A\|h_i - a\|^2_{\Gamma_1} - \|U(h_i - a)\|^2_{\Gamma_0}. \tag{3.7}$$

We note that the same justification as it was stated in Lemma 2.1 is true here for the surrogate functional (3.7). Now by similar derivations to (2.11) but using the adjoint relation (3.6), we can rewrite $\tilde{J}_i^s$ as

$$\begin{aligned}
&\tilde{J}_i^s(\sum_{j=1}^4 h_j, a)\\
=\;& A\|h_i - \Big\{a + \frac{1}{A}U^*\Big(z_0^\delta - U(\sum_{j\neq i} h_j + a)\Big)\Big\}\|^2_{\partial\Omega_i\cap\Gamma_1} + \beta\|\sum_{j=1}^4 h_j\|^2_{\Gamma_1}\\
&+\Big\{\|z_0^\delta - U(\sum_{j\neq i} h_j)\|^2_{\Gamma_0} + A\|a\|^2_{\Gamma_1} - \|U(a)\|^2_{\Gamma_0}\\
&-A\|a + \frac{1}{A}U^*\Big(z_0^\delta - U(\sum_{j\neq i} h_j + a)\Big)\|^2_{\partial\Omega_i\cap\Gamma_1}\Big\},
\end{aligned} \tag{3.8}$$

where $z_0^\delta = z^\delta - u(0)$. We can see that the last term in (3.8) does not depend on h_i, so we can drop it in the minimization of functional $\tilde{J}_i^s$. This suggests us to consider the following local minimization for any $a \in V_i$:

$$\min_{h_i\in V_i} \tilde{J}_i^s(h_i + \sum_{j\neq i} h_j, a) = \min_{h_i\in V_i} A\|h_i - \tilde{z}_i\|^2_{\partial\Omega_i\cap\Gamma_1} + \beta\|\sum_{j=1}^4 h_j\|^2_{\Gamma_1} \tag{3.9}$$

where $\tilde{z}_i$ is given by

$$\tilde{z}_i = a + \frac{1}{A}U^*(z_0^\delta - U(\sum_{j\neq i} h_j + a)). \tag{3.10}$$

This is a quadratic minimization, so we can find its exact minimizer h_i^*:

$$h_i^* = \frac{1}{A+\beta}\Big(A\tilde{z}_i - \beta\sum_{j\neq i} h_j\Big)\Big|_{\partial\Omega_i\cap\Gamma_1}. \tag{3.11}$$

We observe that the minimization (3.9) is completely local, and its solution can be achieved explicitly within the subdomain Ω_i. However, its solution h_i^* needs the data $\tilde{z}_i$ from (3.10), which involves two global solutions of the forward and adjoint systems (3.4) and (3.5), and is clearly not expected in an efficient DD algorithm. Next, we propose some techniques to avoid these two global evaluations so that the resulting DD algorithm involves only local minimizations over the local subdomains. To do so, we introduce two local forward and adjoint operators $U_i(h,p)$ and $U_i^*(\omega,q)$ associated with the global forward and adjoint systems (3.4) and (3.5):

$$\begin{cases} -\nabla\cdot(a(\mathbf{x})\nabla U_i(h,p)) + c(\mathbf{x})U_i(h,p) &= 0 \quad \text{in} \quad \Omega_i, \\ a(\mathbf{x})\frac{\partial U_i(h,p)}{\partial n} &= 0 \quad \text{on} \quad \Gamma_0\cap\partial\Omega_i, \\ a(\mathbf{x})\frac{\partial U_i(h,p)}{\partial n} &= h \quad \text{on} \quad \Gamma_1\cap\partial\Omega_i, \\ U_i(h,p) &= p \quad \text{on} \quad \tilde{\Gamma}_i \end{cases} \tag{3.12}$$

and

$$\begin{cases} -\nabla\cdot(a(\mathbf{x})\nabla U_i^*(\omega,q)) + c(\mathbf{x})U_i^*(\omega,q) &= 0 \quad \text{in} \quad \Omega_i, \\ a(\mathbf{x})\frac{\partial U_i^*(\omega,q)}{\partial n} &= \omega \quad \text{on} \quad \Gamma_0\cap\partial\Omega_i, \\ a(\mathbf{x})\frac{\partial U_i^*(\omega,q)}{\partial n} &= 0 \quad \text{on} \quad \Gamma_1\cap\partial\Omega_i, \\ U_i^*(\omega,q) &= q \quad \text{on} \quad \tilde{\Gamma}_i. \end{cases} \tag{3.13}$$

Using the systems (3.12), (3.13) and the integration by parts formula, we derive the following important relation that will be needed later on:

$$\langle U_i(h,0),\,\omega\rangle_{\Gamma_0\cap\partial\Omega_i} \;=\; \langle h,\,U_i^*(\omega,0)\rangle_{\Gamma_1\cap\partial\Omega_i} \quad \forall\,\omega\in L^2(\Gamma_0\cap\partial\Omega_i). \tag{3.14}$$

By means of the local operators $U_i(h,p)$ in (3.12), we introduce the local functional $J_i(\sum_{j=1}^4 h_j, p)$ for $h_j\in V_j$ $(j=1,2,3,4)$:

$$J_i(\sum_{j=1}^4 h_j,p) \;=\; \|U_i(\sum_{j=1}^4 h_j,p) - z_0^\delta\|^2_{\Gamma_0\cap\partial\Omega_i} + \beta\|\sum_{j=1}^4 h_j\|^2_{\Gamma_1\cap\partial\Omega_i},$$

and a surrogate functional J_i^s for a given $a\in V_i$:

$$\begin{aligned} J_i^s(\sum_{j=1}^4 h_j,p,a) \;=\;& J_i(\sum_{j=1}^4 h_j,p) + A\|h_i-a\|^2_{\Gamma_1\cap\partial\Omega_i} \\ & -\|U_i(h_i-a,0)\|^2_{\Gamma_0\cap\partial\Omega_i}. \end{aligned} \tag{3.15}$$

Using the important fact that $U_i(\sum_{j=1}^4 h_j,p) = U_i(\sum_{j\neq i} h_j,p) + U_i(h_i,0)$ and the adjoint relation (3.14), we can rewrite $J_i^s(\sum_{j=1}^4 h_j,p,a)$ as

$$J_i^s(\sum_{j=1}^4 h_j,p,a)$$

$$
\begin{aligned}
= \quad & A\|h_i - \Big\{a + \frac{1}{A}U_i^*\Big(z_0^\delta - U_i(\sum_{j\neq i} h_j + a, p), 0\Big)\Big\}\|^2_{\Gamma_1\cap\partial\Omega_i} + \beta\|\sum_{j=1}^4 h_j\|^2_{\Gamma_1\cap\partial\Omega_i} \\
& + \Big\{\|z_0^\delta - U_i(\sum_{j\neq i} h_j, p)\|^2_{\Gamma_0\cap\partial\Omega_i} + A\|a\|^2_{\Gamma_1\cap\partial\Omega_i} - \|U_i(a,0)\|^2_{\Gamma_0\cap\partial\Omega_i} \\
& - A\|a + \frac{1}{A}U_i^*\Big(z_0^\delta - U_i(\sum_{j\neq i} h_j + a, p), 0\Big)\|^2_{\Gamma_1\cap\partial\Omega_i}\Big\}. \qquad (3.16)
\end{aligned}
$$

As the last term does not depend on h_i, we are led to the following quadratic minimization:

$$
\min_{h_i\in V_i} J_i^s(\sum_{j=1}^4 h_j, p, a) = \min_{h_i\in V_i} A\|h_i - z_i\|^2_{\Gamma_1\cap\partial\Omega_i} + \beta\|\sum_{j=1}^4 h_j\|^2_{\Gamma_1\cap\partial\Omega_i} \qquad (3.17)
$$

where z_i is given by

$$
z_i = a + \frac{1}{A}U_i^*\Big(z_0^\delta - U_i(\sum_{j\neq i} h_j + a, p), 0\Big).
$$

We can easily find the minimizer to the quadratic optimization (3.17) in explicit form:

$$
\begin{aligned}
h_i^* \quad = \quad & \frac{1}{A+\beta}\Big\{Aa + U_i^*\Big(z_0^\delta - U_i(\sum_{j\neq i} h_j + a, p), 0\Big) \\
& - \beta\sum_{j\neq i} h_j|_{\Gamma_1\cap\partial\Omega_i}\Big\}. \qquad (3.18)
\end{aligned}
$$

As in Section 2.1, we are now ready to formulate two new DD algorithms for the minimization system (3.2) for identifying the heat flux. For the description of the DD algorithms, the same index function $n(\mathbf{x})$ as in (2.21) is used below for any $\mathbf{x}\in\Omega$ and we also introduce an index function for any point $\mathbf{x}\in\Gamma_1$:

$$
\bar{n}(\mathbf{x}) = \Big\{i;\ \mathbf{x}\in\partial\Omega_i\cap\Gamma_1,\ i\in\{1,2,3,4\}\Big\};\quad |\bar{n}(\mathbf{x})| = \text{ number of elements in } \bar{n}(\mathbf{x}).
$$

Algorithm 3.1 (Multiplicative Schwarz Algorithm (MSA)). *Choose a tolerance parameter $\epsilon_1 > 0$, an initial value $h^{(0)} = \sum_{i=1}^4 h_i^{(0)}$ with $h_i^{(0)}\in V_i$ $(i = 1,2,3,4)$, and solve (3.4) for $U(h^{(0)})$; set $p_i^{(0)} := U(h^{(0)})|_{\tilde{\Gamma}_i}$ and $n := 0$.*

1. *Compute $h_i^{(n+1)}\in V_i$ sequentially for $i = 1$ to 4 by*

$$
h_i^{(n+1)} = \text{argmin}_{v_i\in V_i} J_i^s(\sum_{j<i} h_j^{(n+1)} + v_i + \sum_{j>i} h_j^{(n)}, p_i^{(n)}, h_i^{(n)}); \qquad (3.19)
$$

update U_i in Ω_i:

$$
U_i^{(n)} = U_i(\sum_{j\leq i} h_j^{(n+1)} + \sum_{j>i} h_j^{(n)}, p_i^{(n)});
$$

update the inner boundary values on $\tilde{\Gamma}_j$ for $j > i$ if $\tilde{\Gamma}_j\in\Omega_i$:

$$
p_j^{(n)} = U_i^{(n)}|_{\tilde{\Gamma}_j}.
$$

2. *Compute $h^{(n+1)} = \sum_{i=1}^4 h_i^{(n+1)}$.*

3. If $\|h^{(n+1)} - h^{(n)}\|_{\Gamma_1} \le \epsilon_1$, *stop the iteration;*
otherwise update U_i *in subdomains* Ω_i *(*$i = 1, 2, 3, 4$*):*

$$U_i^{(n+1)} = U_i(h^{(n+1)}, p_i^{(n)});$$

update the inner boundary values on $\tilde{\Gamma}_i$ *(*$i = 1, 2, 3, 4$*):*

$$p_i^{(n+1)}(\mathbf{x}) = \frac{1}{|n(\mathbf{x})|} \sum_{j \in n(\mathbf{x})} U_j^{(n+1)}(\mathbf{x}), \quad \forall \mathbf{x} \in \tilde{\Gamma}_i.$$

set $n := n + 1$, *go to Step 1.*

The next algorithm proposes a parallel version of Algorithm 3.1. For this purpose, we introduce a uniform partition of unity $\{\chi_i\}_{i=1}^4$ such that $\sum_{i=1}^4 \chi_i = 1$ and $\|\chi_i\|_\infty \le 1$ and $\text{supp}(\chi_i) \subset \partial\Omega_i \cap \Gamma_1$.

Algorithm 3.2 (Additive Schwarz Algorithm (ASA))**.** *Choose a tolerance parameter* $\epsilon_1 > 0$, *a relaxation parameter* $\lambda \in (0, 1)$, *an initial value* $h^{(0)} = \sum_{i=1}^4 h_i^{(0)}$ *with* $h_i^{(0)} \in V_i$ *(*$i = 1, 2, 3, 4$*), and solve (3.4) for* $U(h^{(0)})$*; set* $p_i^{(0)} := U(h^{(0)})|_{\tilde{\Gamma}_i}$ *and* $n := 0$.

1. Compute $h_i^{(n+1)} \in V_i$ *in parallel for* $i = 1, 2, 3, 4$ *by*

$$h_i^{(n+1)} = \text{argmin}_{v_i \in V_i} J_i^s(\sum_{j \ne i} h_j^{(n)} + v_i, p_i^{(n)}, h_i^{(n)}). \tag{3.20}$$

2. Compute $h^{(n+1)} = \lambda \sum_{i=1}^4 h_i^{(n+1)} + (1 - \lambda) h^{(n)}$.

3. If $\|h^{(n+1)} - h^{(n)}\|_{\Gamma_1} \le \epsilon_1$, *stop the iteration;*
otherwise update U_i *in subdomains* Ω_i *(*$i = 1, 2, 3, 4$*):*

$$U_i^{(n+1)} = U_i(h^{(n+1)}, p_i^{(n)});$$

update the inner boundary values on $\tilde{\Gamma}_i$ *(*$i = 1, 2, 3, 4$*):*

$$p_i^{(n+1)}(\mathbf{x}) = \frac{1}{|n(\mathbf{x})|} \sum_{j \in n(\mathbf{x})} U_j^{(n+1)}(\mathbf{x}) \quad \forall \mathbf{x} \in \tilde{\Gamma}_i.$$

set $h_i^{(n+1)} := \chi_i h^{(n+1)}$, *and* $n := n + 1$, *go to Step 1.*

Remark 3.1. The same as for (3.18), we have explicit expressions for the minimizers $h_i^{(n+1)}$ in (3.19) and (3.20). In our numerical implementations, we will simply take the partition of unity $\{\chi_i\}_{i=1}^4$ used in Algorithm 3.2 as follows:

$$\chi_i(\mathbf{x}) = 1/|\bar{n}(\mathbf{x})| \quad \text{for} \quad \mathbf{x} \in \partial\Omega_i \cap \Gamma_1; \quad \text{and} \quad \chi_i(\mathbf{x}) = 0 \quad \text{for} \quad \mathbf{x} \in \Gamma_1 \backslash \partial\Omega_i.$$

4. Domain decomposition algorithms for the reconstruction of initial temperature.

In this section, we are interested in extending the DD algorithms proposed in sections 2 and 3 for solving the stationary inverse source and flux problems to a time-dependent inverse problem, the identification of the initial temperature in the heat conduction system:

$$\begin{cases} u_t - \nabla \cdot (a(\mathbf{x}) \nabla u) = f(\mathbf{x}, t) & \text{in} \quad \Omega \times (0, T), \\ u = 0 & \text{on} \quad \partial\Omega \times (0, T), \\ u(\mathbf{x}, 0) = \varphi(\mathbf{x}) & \text{in} \quad \Omega. \end{cases} \tag{4.1}$$

We assume that some observation data z^δ of the temperature $u(\mathbf{x}, t)$ are available in Ω or in some small subregion $\omega \subset \Omega$, but with a time history in the range $[T - \sigma, T]$. The inverse problem of our interest is to recover the initial temperature distribution

$\varphi(\mathbf{x})$, using the observation data z^δ. We shall write the solution of system (4.1) as $u(\varphi)$ to emphasize its dependence on the initial temperature $\varphi(\mathbf{x})$.

As described in Section 2, it is easy to verify that $u(\varphi) = U(\varphi) + u(0)$, where $U(\varphi)$ is linear with respect to φ and satisfies the following system

$$\left\{\begin{array}{rcll} U_t - \nabla\cdot(a(\mathbf{x})\nabla U) &=& 0 & \text{in} \quad \Omega\times(0,\,T),\\ U &=& 0 & \text{on} \quad \partial\Omega\times(0,\,T),\\ U(\mathbf{x},0) &=& \varphi(\mathbf{x}) & \text{in} \quad \Omega, \end{array}\right. \tag{4.2}$$

whose variational formulation is given by

$$\int_0^T\int_\Omega U_t\psi d\mathbf{x}dt + \int_0^T\int_\Omega a\nabla U\cdot\nabla\psi d\mathbf{x}dt = 0 \quad \forall\,\psi\in L^2(0,T;H_0^1(\Omega))\,. \tag{4.3}$$

Let $z_0^\delta = z^\delta - u(0)$, then we can formulate our inverse problem as the following regularized output least-squares minimization:

$$\min_{\varphi\in L^2(\Omega)} J(\varphi) = \min_{\varphi\in L^2(\Omega)} \int_{T-\sigma}^T \|U(\varphi) - z_0^\delta\|^2_{L^2(\Omega)}dt + \beta\|\varphi\|^2_{L^2(\Omega)}. \tag{4.4}$$

Now we introduce the adjoint system of the forward problem (4.2):

$$\left\{\begin{array}{rcll} U_t^* + \nabla\cdot(a(\mathbf{x})\nabla U^*) &=& 0 & \text{in} \quad \Omega\times(0,\,T),\\ U^* &=& 0 & \text{on} \quad \partial\Omega\times(0,\,T),\\ U^*(\mathbf{x},T) &=& \omega & \text{in} \quad \Omega, \end{array}\right. \tag{4.5}$$

which is linear with respect to ω. Next we derive a very useful relation:

$$\langle U(\varphi)(\cdot,t),\,\omega\rangle_{L^2(\Omega)} = \langle\varphi,\,U^*(\omega)(\cdot,T-t)\rangle_{L^2(\Omega)} \quad \forall\, t\in[0,T]\,. \tag{4.6}$$

Clearly, this is true for $t = 0$ by the initial and terminal conditions in (4.2) and (4.5). To verify it for $t\in(0,T]$, we define $U^{*,s}(\omega)$ for $s\in(0,T]$:

$$\left\{\begin{array}{rcll} U_t^{*,s} + \nabla\cdot(a(\mathbf{x})\nabla U^{*,s}) &=& 0 & \text{in} \quad \Omega\times(0,\,s),\\ U^{*,s} &=& 0 & \text{on} \quad \partial\Omega\times(0,\,s),\\ U^{*,s}(\mathbf{x},s) &=& \omega & \text{in} \quad \Omega. \end{array}\right. \tag{4.7}$$

It is easy to find the following relation,

$$U^*(\omega)(\mathbf{x},T-s) = U^{*,s}(\omega)(\mathbf{x},0), \tag{4.8}$$

and the variational formulation of (4.7),

$$-\int_0^s\int_\Omega U_t^{*,s}\psi d\mathbf{x}dt + \int_0^s\int_\Omega a\nabla U^{*,s}\cdot\nabla\psi d\mathbf{x}dt = 0 \quad \forall\,\psi\in L^2(0,s;H_0^1(\Omega)). \tag{4.9}$$

Using $U^{*,s}$ in (4.7) and its property (4.8), we see (4.6) immediately from the following relation

$$\langle U(\varphi)(\cdot,s),\,\omega\rangle_{L^2(\Omega)} = \langle\varphi,\,U^{*,s}(\omega)(\cdot,0)\rangle_{L^2(\Omega)}\,. \tag{4.10}$$

To check this relation, we use (4.3) with the terminal time T replaced by s, then take $\psi = U^{*,s}$ and integrate by parts with respect to t to obtain

$$\begin{aligned} &\int_\Omega U(\mathbf{x},s)U^{*,s}(\mathbf{x},s)d\mathbf{x} - \int_\Omega U(\mathbf{x},0)U^{*,s}(\mathbf{x},0)d\mathbf{x} - \int_0^s\int_\Omega UU_t^{*,s}d\mathbf{x}dt\\ &+ \int_0^s\int_\Omega a(\mathbf{x})\nabla U\cdot\nabla U^{*,s}d\mathbf{x}dt = 0. \end{aligned} \tag{4.11}$$

Now the desired relation (4.10) follows readily from the initial and terminal conditions in (4.2) and (4.7) and equation (4.9) with $\psi = U$.

Next we shall follow sections 2 and 3 to derive some overlapping domain decomposition method for solving the time-dependent minimization (4.4). As in section 2.1, Ω is divided into the overlapping subdomains Ω_i $(i = 1, 2, 3, 4)$, and the feasible constraint space $L^2(\Omega)$ can be decomposed accordingly into the following subspaces:

$$V_i = \Big\{\varphi \in L^2(\Omega);\ \mathrm{supp}(\varphi) \subset \Omega_i\Big\}, \quad i = 1,\ 2,\ 3,\ 4.$$

In order to avoid any global solution of the forward and adjoint systems (4.2) and (4.5) in our DD algorithms, we introduce their local variants, namely, the solutions $U_i(\varphi, p)$ and $U_i^*(\omega, p)$ to the following systems:

$$\begin{cases} U_i(\varphi,p)(\mathbf{x},t)_t - \nabla\cdot(a\nabla U_i(\varphi,p)(\mathbf{x},t)) &= 0 \quad \text{in} \quad \Omega_i\times(0,\,T),\\ U_i(\varphi,p)(\mathbf{x},t) &= 0 \quad \text{on} \quad (\partial\Omega\cap\partial\Omega_i)\times(0,\,T),\\ U_i(\varphi,p)(\mathbf{x},t) &= p \quad \text{on} \quad \tilde{\Gamma}_i\times(0,\,T),\\ U_i(\varphi,p)(\mathbf{x},0) &= \varphi \quad \text{in} \quad \Omega_i \end{cases}$$

and

$$\begin{cases} U_i^*(\omega,p)(\mathbf{x},t)_t + \nabla\cdot(a\nabla U_i^*(\omega,p)(\mathbf{x},t)) &= 0 \quad \text{in} \quad \Omega_i\times(0,\,T),\\ U_i^*(\omega,p)(\mathbf{x},t) &= 0 \quad \text{on} \quad (\partial\Omega\cap\partial\Omega_i)\times(0,\,T),\\ U_i^*(\omega,p)(\mathbf{x},t) &= p \quad \text{on} \quad \tilde{\Gamma}_i\times(0,\,T),\\ U_i^*(\omega,p)(\mathbf{x},T) &= \omega \quad \text{in} \quad \Omega_i. \end{cases}$$

Noting that $U_i(\varphi, 0) = U_i^*(\omega, 0) = 0$ on $\partial\Omega_i$, we can derive as we did for (4.6) that

$$\langle U_i(\varphi,0)(\cdot,t),\,\omega\rangle_{L^2(\Omega_i)} = \langle\varphi,\,U_i^*(\omega,0)(\cdot,T-t)\rangle_{L^2(\Omega_i)}\,. \tag{4.12}$$

Now we can define a local functional $J_i(\sum_{j=1}^4\varphi_j, p)$ for $\varphi_j \in V_j$ $(j = 1, 2, 3, 4)$:

$$J_i(\sum_{j=1}^4\varphi_j,p) = \int_{T-\sigma}^T \|U_i(\sum_{j=1}^4\varphi_j,p)(\cdot,t) - z_0^\delta\|^2_{L^2(\Omega_i)}dt + \beta\|\sum_{j=1}^4\varphi_j\|^2_{L^2(\Omega_i)}$$

and introduce a surrogate functional J_i^s for any $a \in V_i$:

$$\begin{aligned} J_i^s(\sum_{j=1}^4\varphi_j,p,a) &= J_i(\sum_{j=1}^4\varphi_j,p) + A\sigma\|\varphi_i - a\|^2_{L^2(\Omega_i)}\\ &\quad - \int_{T-\sigma}^T \|U_i(\varphi_i - a,0)(\cdot,t)\|^2_{L^2(\Omega_i)}dt. \end{aligned}$$

We note that the same justification as it was stated in Lemma 2.1 is true here for the above surrogate functional. Now using the fact that $U_i(\sum_{j=1}^4\varphi_j,p) = U_i(\sum_{j\neq i}\varphi_j,p) + U_i(\varphi_i,0)$ and the adjoint relation (4.12), we can rewrite

$$\begin{aligned} &J_i^s(\sum_{j=1}^4\varphi_j,p,a)\\ =\ &\int_{T-\sigma}^T\{\|U_i(\varphi_i,0)(\cdot,t)\|^2_{\Omega_i} - 2\langle U_i(\varphi_i,0)(\cdot,t),\,z_0^\delta - U_i(\sum_{j\neq i}\varphi_j,p)(\cdot,t)\rangle_{\Omega_i}\\ &+\|z_0^\delta - U_i(\sum_{j\neq i}\varphi_j,p)(\cdot,t)\|^2_{\Omega_i}\}dt + \beta\|\sum_{j=1}^4\varphi_j\|^2_{\Omega_i} + A\sigma\langle\varphi_i,\,\varphi_i - 2a\rangle_{\Omega_i} \end{aligned}$$

$$
\begin{aligned}
&+A\sigma\|a\|^2_{\Omega_i} - \int_{T-\sigma}^{T} \{\|U_i(\varphi_i,0)(\cdot,t)\|^2_{\Omega_i} - 2\langle U_i(\varphi_i,0)(\cdot,t),\, U_i(a,0)(\cdot,t)\rangle_{\Omega_i} \\
&+\|U_i(a,0)(\cdot,t)\|^2_{\Omega_i}\}dt \\
= \;& A\sigma\langle \varphi_i,\, \varphi_i - 2\{a + \frac{1}{A\sigma}\int_{T-\sigma}^{T} U_i^*(z_0^\delta - U_i(\sum_{j\neq i}\varphi_j + a, p)(\cdot,t),0)(\cdot,T-t)dt\}\rangle_{\Omega_i} \\
&+\beta\|\sum_{j=1}^{4}\varphi_j\|^2_{\Omega_i} + \int_{T-\sigma}^{T}\{\|z_0^\delta - U_i(\sum_{j\neq i}\varphi_j, p)(\cdot,t)\|^2_{\Omega_i} - \|U_i(a,0)(\cdot,t)\|^2_{\Omega_i}\}dt \\
&+A\sigma\|a\|^2_{\Omega_i} \\
= \;& A\sigma\|\varphi_i - \{a + \frac{1}{A\sigma}\int_{T-\sigma}^{T} U_i^*(z_0^\delta - U_i(\sum_{j\neq i}\varphi_j + a, p)(\cdot,t),0)(\cdot,T-t)dt\}\|^2_{\Omega_i} \\
&+\beta\|\sum_{j=1}^{4}\varphi_j\|^2_{\Omega_i} + \Big\{\int_{T-\sigma}^{T}\{\|z_0^\delta - U_i(\sum_{j\neq i}\varphi_j, p)(\cdot,t)\|^2_{\Omega_i} - \|U_i(a,0)(\cdot,t)\|^2_{\Omega_i}\}dt \\
&-A\sigma\|a + \frac{1}{A\sigma}\int_{T-\sigma}^{T} U_i^*(z_0^\delta - U_i(\sum_{j\neq i}\varphi_j + a, p)(\cdot,t),0)(\cdot,T-t)dt\|^2_{\Omega_i} \\
&+A\sigma\|a\|^2_{\Omega_i}\Big\}. \qquad (4.13)
\end{aligned}
$$

We can easily see that the last term above does not depend on φ_i, so it will not affect the local minimization over Ω_i if we drop the term in the functional J_i^s. This leads us to consider the following functional for a given $a \in V_i$:

$$
\min_{\varphi_i\in V_i} J_i^s(\sum_{j=1}^{4}\varphi_j, p, a) = \min_{\varphi_i\in V_i} A\sigma\|\varphi_i - z_i\|^2_{\Omega_i} + \beta\|\sum_{j=1}^{4}\varphi_j\|^2_{\Omega_i}, \qquad (4.14)
$$

where z_i is given by

$$
z_i = a + \frac{1}{A\sigma}\int_{T-\sigma}^{T} U_i^*(z_0^\delta - U_i(\sum_{j\neq i}\varphi_j + a, p)(\cdot,t),0)(\cdot,T-t)dt.
$$

Clearly the minimization (4.14) is quadratic, so we can find its exact minimizer φ_i^*:

$$
\begin{aligned}
\varphi_i^* \;=\; & \frac{1}{A\sigma+\beta}\Big\{A\sigma a + \int_{T-\sigma}^{T} U_i^*(z_0^\delta - U_i(\sum_{j\neq i}\varphi_j + a, p)(\cdot,t),0)(\cdot,T-t)dt \\
& -\beta\sum_{j\neq i}\varphi_j|_{\Omega_i}\Big\}. \qquad (4.15)
\end{aligned}
$$

By means of the local minimizations (4.14), we are now ready to formulate two new DD algorithms for solving the minimization (4.4) for the reconstruction of the initial temperature. The same index function $n(\mathbf{x})$ as in (2.21) is used below for any $\mathbf{x} \in \Omega$.

Algorithm 4.1 (Multiplicative Schwarz Algorithm (MSA)). *Choose a tolerance parameter $\epsilon_1 > 0$, an initial value $\varphi^{(0)} = \sum_{i=1}^{4}\varphi_i^{(0)}$ with $\varphi_i^{(0)} \in V_i$ $(i = 1,2,3,4)$, and solve (4.2) for $U(\varphi^{(0)})$; set $p_i^{(0)} := U(\varphi^{(0)})|_{\tilde{\Gamma}_i}$ and $n := 0$.*

1. *Compute* $\varphi_i^{(n+1)} \in V_i$ *sequentially for* $i = 1$ *to* 4 *by*

$$\varphi_i^{(n+1)} = \text{argmin}_{v_i \in V_i} J_i^s(\sum_{j<i} \varphi_j^{(n+1)} + v_i + \sum_{j>i} \varphi_j^{(n)}, p_i^{(n)}, \varphi_i^{(n)}); \tag{4.16}$$

update U_i *in* Ω_i*:*

$$U_i^{(n)} = U_i(\sum_{j\le i} \varphi_j^{(n+1)} + \sum_{j>i} \varphi_j^{(n)}, p_i^{(n)});$$

update the inner boundary values on $\tilde{\Gamma}_j$ *for* $j > i$ *if* $\tilde{\Gamma}_j \in \Omega_i$*:*

$$p_j^{(n)} = U_i^{(n)}|_{\tilde{\Gamma}_j}.$$

2. *Compute* $\varphi^{(n+1)} = \sum_{i=1}^4 \varphi_i^{(n+1)}$.
3. *If* $\|\varphi^{(n+1)} - \varphi^{(n)}\|_\Omega \le \epsilon_1$*, stop the iteration;*
   otherwise update U_i *in subdomain* Ω_i *(*$i = 1, 2, 3, 4$*):*

$$U_i^{(n+1)} = U_i(\varphi^{(n+1)}, p_i^{(n)});$$

update the inner boundary values on $\tilde{\Gamma}_i$ *(*$i = 1, 2, 3, 4$*):*

$$p_i^{(n+1)}(\mathbf{x}) = \frac{1}{|n(\mathbf{x})|} \sum_{j\in n(\mathbf{x})} U_j^{(n+1)}(\mathbf{x}), \quad \forall\, \mathbf{x} \in \tilde{\Gamma}_i.$$

set $n := n + 1$*, go to Step 1.*

The next algorithm is a parallel version of Algorithm 4.1.

Algorithm 4.2 (Additive Schwarz Algorithm (ASA))**.** *Choose a tolerance parameter* $\epsilon_1 > 0$*, a relaxation parameter* $\lambda \in (0, 1)$*, an initial value* $\varphi^{(0)} = \sum_{i=1}^4 \varphi_i^{(0)}$ *with* $\varphi_i^{(0)} \in V_i$ *(*$i = 1, 2, 3, 4$*), and solve (4.2) for* $U(\varphi^{(0)})$*; set* $p_i^{(0)} := U(\varphi^{(0)})|_{\tilde{\Gamma}_i}$ *and* $n := 0$.

1. *Compute* $\varphi_i^{(n+1)} \in V_i$ *in parallel for* $i = 1, 2, 3, 4$ *by*

$$\varphi_i^{(n+1)} = \text{argmin}_{v_i \in V_i} J_i^s(\sum_{j\ne i} \varphi_j^{(n)} + v_i, p_i^{(n)}, \varphi_i^{(n)}). \tag{4.17}$$

2. *Compute* $\varphi^{(n+1)} = \lambda \sum_{i=1}^4 \varphi_i^{(n+1)} + (1 - \lambda)\varphi^{(n)}$.
3. *If* $\|\varphi^{(n+1)} - \varphi^{(n)}\|_\Omega \le \epsilon_1$*, stop the iteration;*
   otherwise update U_i *in subdomains* Ω_i *(*$i = 1, 2, 3, 4$*):*

$$U_i^{(n+1)} = U_i(\varphi^{(n+1)}, p_i^{(n)});$$

update the inner boundary values on $\tilde{\Gamma}_i$ *(*$i = 1, 2, 3, 4$*):*

$$p_i^{(n+1)}(\mathbf{x}) = \frac{1}{|n(\mathbf{x})|} \sum_{j\in n(\mathbf{x})} U_j^{(n+1)}(\mathbf{x}) \quad \forall\, \mathbf{x} \in \tilde{\Gamma}_i.$$

set $\varphi_i^{(n+1)} := \chi_i \varphi^{(n+1)}$*, and* $n := n + 1$*, go to Step 1.*

Remark 4.1. The same as for (4.15), we have explicit expressions for the minimizers $\varphi_i^{(n+1)}$ in (4.16) and (4.17).

5. **Numerical experiments.** In this section, we shall apply the DD algorithms that were proposed in the previous Sections 2-4 to identify the source strength in the elliptic system (2.1), the heat flux in the system (3.1) and the initial temperature in the parabolic system (4.1) respectively.

We choose the domain $\Omega = (0,1) \times (0,2)$ and decompose it into four overlapping subdomains: $\Omega_1 = (0,4/7) \times (6/7,2)$, $\Omega_2 = (3/7,1) \times (6/7,2)$, $\Omega_3 = (0,4/7) \times (0,8/7)$, $\Omega_4 = (3/7,1) \times (0,8/7)$. Then we triangulate domain Ω into $N \times M$ small squares of equal size and further divide each square through its diagonal into two triangles. This results in a finite element triangulation of domain Ω, which is done in such a way that it is consistent with the subdomain decompositions. All the elliptic problems involved in DD algorithms are solved by the continuous linear finite element method, while all the parabolic problems are solved by the continuous linear finite element method in space and the Crank-Nicolson scheme in time.

The parameters involved in the DD algorithms are chosen as follows. The initial guesses are set to be identically equal to some constants, which as we see are rather poor initial guesses for all our testing inverse problems. We take the relaxation parameter $\lambda = 1/2$ in all the numerical experiments. The noisy data z^δ is obtained by adding some uniform random noise to the exact data, i.e., $z^\delta = u + \delta R\, u$, where R is a uniform random function varying in the range [-1,1].

Next, we discuss about some appropriate tools we should use to better measure the convergence of the DDMs. We take the first inverse problem of recovering the source strength in the elliptic system (2.1) as an example. The exact source strength f in (2.1) is approximated through the minimisation problem (2.6), which is solved by the new DDMs, namely Algorithms 2.1 and 2.2. We solve the finite element discretised system of the minimisation problem (2.6) by the regularized Landweber iteration, and write by f_i^* the finite element minimiser corresponding to a mesh T_{h_i} on domain Ω, and by $f_i^{(k)}$ the approximate solution of f_i^* obtained by the kth iteration of the DDMs.

For a direct problem, the exact solution is usually obtained or approximated using a sufficiently fine mesh. But based on the general regularisation theory [8] and convergence of discrete solutions of inverse problems [15], an exact solution achieved with a sufficiently fine mesh is often not the desired approximate solution one should have in most applications, as such a solution is not the approximate solution with the best accuracy. After many numerical tests, we find that the finite element discrete solution f_i^* with the mesh $T_{h_i} = 28 \times 56$ gives the "best" approximate solution when the noise level is $\delta = 2\%$. Based on this observation, we first apply the regularized Landweber iteration to compute the discrete minimiser f_i^* for three nested meshes $T_{h_i} = (7 \times 2^{i-1}) \times (14 \times 2^{i-1})$, $i = 1,2,3$. Then in order to test the convergence of DDMs, for each mesh T_{h_i} we apply the newly proposed DDMs to compute f_i^* and record the approximate solution $f_i^{(k)}$ of f_i^* obtained by the kth iteration of the DDMs. For the noise level $\delta = 2\%$, DDMs will be terminated when the relative L^2-norm error reaches 0.08, namely $\|f_i^{(k)} - f_i^*\|/\|f_i^*\| \le 0.08$, and the corresponding numbers of DDM iterations are then listed for each numerical example.

In all the subsequent numerical results, the same measurements for testing the convergence of DDMs as we have discussed above for the inverse source strength in system (2.1) are applied for the inverse problems of recovering the heat flux in system (3.1) and the initial temperature in the parabolic system (4.1).

TABLE 5.1. Number of iterations by MSA and ASA for Example 5.1

Algorithm	N	M	β	k
MSA	7	14	10^{-5}	10
	14	28	10^{-5}	13
	28	56	10^{-5}	16
ASA	7	14	10^{-5}	19
	14	28	10^{-5}	26
	28	56	10^{-5}	31

We first show three numerical examples for reconstructing the source strength $f(\mathbf{x})$ in the system (2.1), with $a(\mathbf{x}) = (x+y)/100$, $c(\mathbf{x}) = 1$ in Ω and $g(\mathbf{x}) = 0$ on $\partial\Omega$. We take a constant initial guess $f^{(0)} = 0$ in Ω and the noise level $\delta = 2\%$ and $A = 1$.

Example 5.1. We take the exact source strength $f = \sin(2\pi x)\sin(2\pi y)$.

Figure 5.1 shows the exact and numerically recovered source strengths, while Table 5.1 gives the number of iterations by Algorithms 2.1 and 2.2 with the relative L^2-norm error being 0.08.

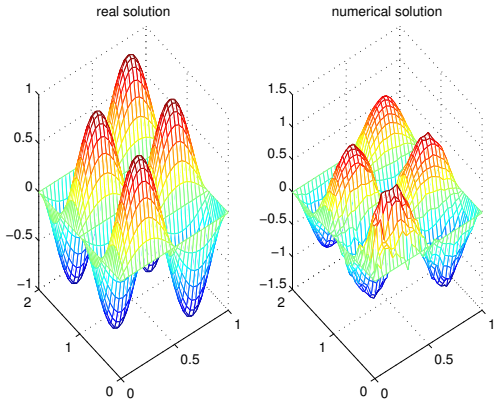

FIGURE 5.1. Exact and numerically recovered source strengths for Example 5.1

Example 5.2. We take the exact source strength $f = 2\sin(2\pi x)y(y-1)(y-2)$.

Figure 5.2 shows the exact and numerically recovered source strengths, while Table 5.2 gives the number of iterations by Algorithms 2.1 and 2.2 with the relative L^2-norm error being 0.08.

Example 5.3. We take the exact source strength $f = 10y\sin(2\pi y)x(x-1/2)(x-1)$.

Figure 5.3 shows the exact and numerically recovered source strengths, while Table 5.3 gives the number of iterations by Algorithms 2.1 and 2.2 with the relative L^2-norm error being 0.08, and Table 5.4 shows the relative L^2-norm errors achieved with different regularisation parameters β for the mesh 14×28 by MSA and ASA respectively.

Table 5.2. Number of iterations by MSA and ASA for Example 5.2

Algorithm	N	M	β	k
MSA	7	14	10^{-5}	8
	14	28	10^{-5}	13
	28	56	10^{-5}	16
ASA	7	14	10^{-5}	13
	14	28	10^{-5}	24
	28	56	10^{-5}	30

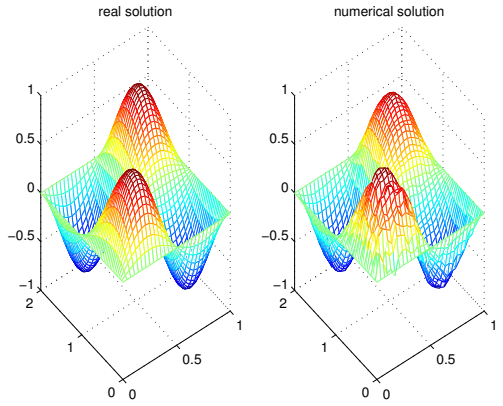

Figure 5.2. Exact and numerically recovered source strengths for Example 5.2

Table 5.3. Number of iterations by MSA and ASA for Example 5.3

Algorithm	N	M	β	k
MSA	7	14	10^{-5}	11
	14	28	10^{-5}	14
	28	56	10^{-5}	16
ASA	7	14	10^{-5}	23
	14	28	10^{-5}	29
	28	56	10^{-5}	34

We can see from Figures 5.1-5.3 that the numerical reconstructed source strengths, with a 2% noise in the data, appear to be quite satisfactory, in view of the severe ill-posedness of the inverse source problem and the complicated profiles of the exact source strengths, especially in Example 5.1, where the source strength oscillates frequently between 8 peaks and valleys. More importantly, we observe from Tables 5.1-5.3 that the convergence of the DD algorithms are rather reasonable: the number of iterations grows only mildly with the mesh refinement.

In addition, we observe from Table 5.4 that the DDMs are rather robust with respect to the choice of the regularisation parameter β: the accuracy of the numerical reconstruction does not change much when the parameter is taken in a specific range of large scale.

TABLE 5.4. L^2-norm errors with different β by MSA & ASA for Example 5.3

Algorithm	β	error	k
MSA	10^{-3}	0.0568	18
	10^{-4}	0.0590	18
	10^{-5}	0.0592	18
	10^{-6}	0.0592	18
	10^{-7}	0.0592	18
ASA	10^{-3}	0.0570	37
	10^{-4}	0.0589	37
	10^{-5}	0.0591	37
	10^{-6}	0.0592	37
	10^{-7}	0.0592	37

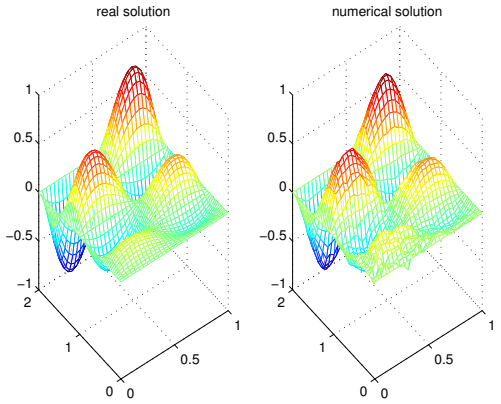

FIGURE 5.3. Exact and numerically recovered source strengths for Example 5.3

Next, we demonstrate three numerical examples for the reconstructions of the initial temperature in the heat conductive system (4.1), by two DD algorithms, namely Algorithms 4.1 and 4.2 proposed in Section 4. In our experiments, we take $A = 1$, the noise level $\delta = 2\%$, $a(\mathbf{x}) = 1$, $f(\mathbf{x}, t) = 0$, the terminal time $T = 4$, and the constant initial guess $\varphi^{(0)} = 0$.

Example 5.4. We take the exact initial temperature $\varphi = \sin(2\pi x)\sin(2\pi y)$.

Figure 5.4 shows the exact and numerically recovered initial temperature, while Table 5.5 gives the number of iterations by Algorithms 4.1 and 4.2 with the relative L^2-norm error being 0.08.

Example 5.5. We take the exact initial temperature $\varphi = 2\sin(2\pi x)y(y-1)(y-2)$.

Figure 5.5 shows the exact and numerically recovered initial temperature, while Table 5.6 gives the number of iterations by Algorithms 4.1 and 4.2 with the relative L^2-norm error being 0.08.

Example 5.6. We take the exact initial temperature $\varphi = 10y\sin(2\pi y)x(x - 1/2)(x-1)$.

Table 5.5. Number of iterations by MSA and ASA for Example 5.4

Algorithm	N	M	β	k
MSA	7	14	$5 * 10^{-5}$	7
	14	28	$5 * 10^{-5}$	9
	28	56	$5 * 10^{-5}$	9
ASA	7	14	$5 * 10^{-5}$	16
	14	28	$5 * 10^{-5}$	19
	28	56	$5 * 10^{-5}$	20

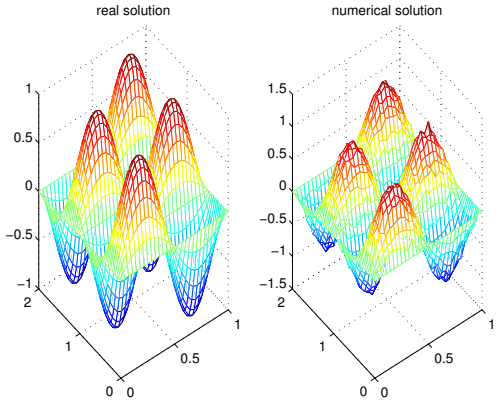

Figure 5.4. Exact and numerically recovered initial temperatures for Example 5.4

Table 5.6. Number of iterations by MSA and ASA for Example 5.5

Algorithm	N	M	β	k
MSA	7	14	$5 * 10^{-5}$	20
	14	28	$5 * 10^{-5}$	23
	28	56	$5 * 10^{-5}$	25
ASA	7	14	$5 * 10^{-5}$	41
	14	28	$5 * 10^{-5}$	48
	28	56	$5 * 10^{-5}$	50

Figure 5.6 shows the exact and numerically recovered initial temperature, while Table 5.7 gives the number of iterations by Algorithms 4.1 and 4.2 with the relative L^2-norm error being 0.08.

We can see from Figures 5.4-5.6 that the numerical reconstructed initial temperatures, with a 2% noise in the data, appear to be quite satisfactory, in view of the severe ill-posedness of the inverse initial temperature problem and the complicated profiles of the exact initial temperatures, especially in Example 5.4 where the initial temperature oscillates frequently between 8 peaks and valleys. More importantly, we observe from Tables 5.5-5.7 that the convergence of the DD algorithms are rather reasonable: the number of iterations grows still mildly with the mesh refinement. These important features of the algorithms are also observed in other examples of this section.

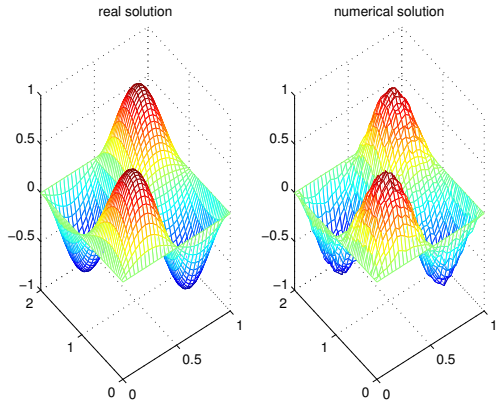

FIGURE 5.5. Exact and numerically recovered initial temperatures for Example 5.5

TABLE 5.7. Number of iterations by MSA and ASA for Example 5.6

Algorithm	N	M	β	k
MSA	7	14	$5 * 10^{-5}$	7
	14	28	$5 * 10^{-5}$	9
	28	56	$5 * 10^{-5}$	10
ASA	7	14	$5 * 10^{-5}$	17
	14	28	$5 * 10^{-5}$	20
	28	56	$5 * 10^{-5}$	21

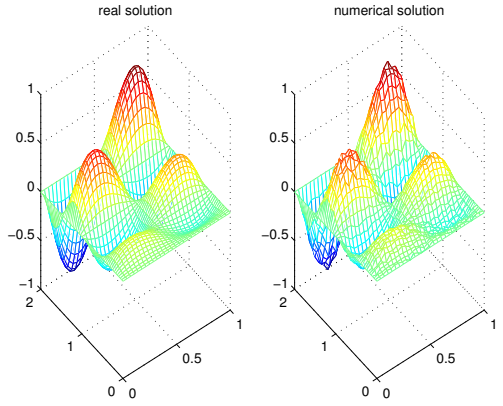

FIGURE 5.6. Exact and numerically recovered initial temperatures for Example 5.6

Finally, We present a numerical test for the reconstruction of fluxes on the partial boundary $\Gamma_1 = \{(x, y);\ x = 1,\ 0 \le y \le 2\}$ in the system (3.1), where we take $g(\mathbf{x}) = 0$ on Γ_0, $f(\mathbf{x}) = 0$ and $a(\mathbf{x}) = c(\mathbf{x}) = 1$ in Ω.

Table 5.8. Number of iterations by MSA and ASA for Example 5.7

Algorithm	N	M	β	k
MSA	7	14	10^{-5}	27
	14	28	10^{-5}	19
	28	56	10^{-5}	18
ASA	7	14	10^{-5}	52
	14	28	10^{-5}	42
	28	56	10^{-5}	41

Example 5.7. We take the exact flux $h(x, y) = (y-1)^2 + 1$ on Γ_1 with a constant initial guess $h^{(0)} = 1$, $\delta = 2\%$ and $A = 5$.

Figure 5.7 shows the exact and numerically recovered heat flux, while Table 5.8 gives the number of iterations by Algorithms 3.1 and 3.2 with the relative L^2-norm error being 0.08.

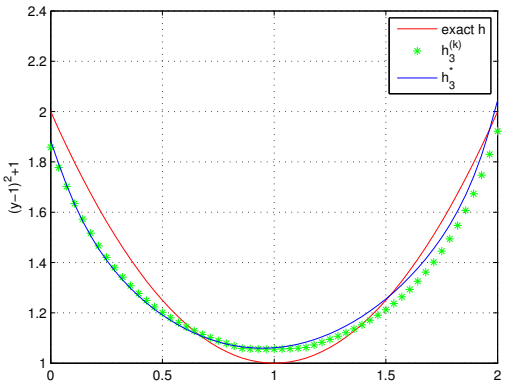

Figure 5.7. Exact and numerically recovered fluxes for Example 5.7

We can see from Figure 5.7 that the numerical reconstructed fluxes, with a 2% noise in the data, appear to be quite satisfactory, in view of the severe ill-posedness of the inverse flux problem. More importantly, we observe that the algorithms converge globally, starting with a rather poor initial guess, and its convergence behaves very well (see Table 5.8): the number of iterations grows mildly with the mesh refinement except when the mesh is too coarse.

6. **Concluding remarks.** We have proposed in this work several overlapping domain decomposition algorithms for solving some representative linear inverse problems, including the identification of the fluxes, the source intensity and the initial temperature in second order elliptic and parabolic systems. The algorithms are constructed in a way that only small sub-minimizations are needed to solve on the subdomains of the original global domain at each iteration. The algorithms can be realised easily and very efficiently, with explicit solutions for all the sub-minimizations involved. And we have observed from many numerical examples that the algorithms converge globally, and converge with rather poor initial guesses. More importantly,

the convergences of these algorithms do not deteriorate or deteriorate only mildly with the refinement of finite element mesh.

Our future work includes the extension of the proposed overlapping domain decomposition algorithms to nonlinear inverse problems, such as the constructions of the diffusivity coefficient, the radiative coefficient and Robin coefficient in elliptic and parabolic systems.

Acknowledgments. The authors would like to thank two anonymous referees for their many constructive comments and suggestions, which have helped us improve the organisation and the quality of the paper essentially.

REFERENCES

[1] R. C. Aster, B. Borchers and C. H. Thurber, *Parameter Estimation and Inverse Problems*, Elsevier Academic Press, New York, 2005.
[2] H. T. Banks and K. Kunisch, *Estimation Techniques for Distributed Parameter Systems*, Birkhauser, Boston, 1989.
[3] X. Cai, S. Liu and J. Zou, Parallel overlapping domain decomposition methods for coupled inverse elliptic problems, *Comm. Appl. Math. Comput. Sci.*, **4** (2009), 1–26.
[4] T. Chan and T. Mathew, Domain decomposition algorithms, *Acta Numerica*, (1994), 61–143.
[5] T. Chan and X. Tai, Identification of discontinuous coefficients from elliptic problems using total variation regularization, *SIAM J. Sci. Comput.*, **25** (2003), 881–904.
[6] H. Chang and D. Yang, A Schwarz domain decomposition method with gradient projection for optimal control governed by elliptic partial differential equations, *J. Comput. Appl. Math.*, **235** (2011), 5078–5094.
[7] I. Daubechies, M. Defrise and C. Demol, An iterative thresholding algorithm for linear inverse problems, *Comm. Pure Appl. Math.*, **57** (2004), 1413–1457.
[8] H. Engl, M. Hanke and A. Neubauer, *Regularization of Inverse Problems*, Kluwer Academic Publishers, The Netherlands, 2000.
[9] M. Heinkenschloss and M. Herty, A spatial domain decomposition method for parabolic optimal control problems, *J. Comput. Appl. Math.*, **201** (2007), 88–111.
[10] M. Heinkenschloss and H. Nguyen, Neumann-Neumann domain decomposition preconditioners for linear-quadratic elliptic optimal control problems, *SIAM Journal on Scientific Computing*, **28** (2006), 1001–1028.
[11] K. Ito and J. Zou, Identification of some source densities of the distribution type, *J. Comput. Appl. Math.*, **132** (2001), 295–308.
[12] J. Li and J. Zou, A multilevel model correction method for parameter identification, *Inverse Problems*, **23** (2007), 1759–1786.
[13] X. Tai, J. Froyen, M. Espedal and T. Chan, Overlapping domain decomposition and multigrid methods for inverse problems, *Contemporary Mathematics*, **218** (1998), 523–529.
[14] A. Toselli and O. Widlund, *Domain Decomposition Methods–Algorithms and Theory*, Springer-Verlag, New York, 2004.
[15] L. Wang and J. Zou, Error estimates of finite element methods for parameter identifications in elliptic and parabolic systems, *Disc. Cont. Dynam. Sys., Series B*, **14** (2010), 1641–1670.
[16] J. Xie and J. Zou, Numerical reconstruction of heat fluxes, *SIAM J. Numer. Anal.*, **43** (2005), 1504–1535.
[17] J. Xu, Iterative methods by space decomposition and subspace correction, *SIAM Review*, **34** (1992), 581–613.
[18] J. Xu and J. Zou, Some nonoverlapping domain decomposition methods, *SIAM Review*, **40** (1998), 857–914.

Received October 2013; revised May 2014.

E-mail address: jiangdaijun@mail.ccnu.edu.cn
E-mail address: hfeng.math@whu.edu.cn
E-mail address: zou@math.cuhk.edu.hk