Computers and Mathematics with Applications 111 (2022) 98–108

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

www.elsevier.com/locate/camwa

An efficient two-level overlapping domain decomposition method for recovering unsteady sources of 3D parabolic problems

Xiaomao Deng [a], Zi-Ju Liao [b,*], Xiao-Chuan Cai [c]

[a] School of Mathematics and Statistics, Guangdong University of Foreign Studies, Guangzhou 510006, China
[b] Department of Mathematics, Jinan University, Guangzhou 510632, China
[c] Department of Mathematics, University of Macau, Macau, China

ARTICLE INFO

Keywords:
Domain decomposition
Parallel computing
Source identification
Inverse problems

ABSTRACT

We develop a parallel two-level domain decomposition method for the 3D unsteady source identification problem governed by a parabolic partial differential equation (PDE). The domain of the PDE is firstly decomposed into several overlapping subdomains and the original inverse source identification problem is then transformed into smaller independent subproblems defined on these subdomains. Each subproblem is formulated as a PDE-constrained optimization problem with appropriate conditions prescribed on the inner boundaries and discretized by finite element method. The resulting coupled algebraic systems are solved simultaneously by restarted GMRES method with a space-time restricted additive Schwarz preconditioner. When forming the preconditioner, a second level of domain decomposition is introduced for each subdomain. The solutions of these subproblems are combined together to form an approximated global solution to the original inverse problem by discarding the overlapping parts of the solution. Since all the subproblems are solved independently, the two-level domain decomposition method provides higher degree of parallelism and saves much computing time. Numerical experiments conducted on a supercomputer with thousands of processor cores validate the efficiency and robustness of the proposed approach.

1. Introduction

The parabolic convection-diffusion equation describes how the physical quantities such as mass, energy, momentum and concentration are transferred with the convection and diffusion processes of the fluid. It is widely used in the simulation of hydrodynamics, heat transfer, chemical reaction, environmental pollution or viral aerosol transmission, etc. The canonical form of the convection-diffusion equation is $\frac{\partial u}{\partial t} - \nabla \cdot (a\nabla u) + \nabla \cdot (\mathbf{v}u) = f$, where a is the diffusion parameter, the vector $\mathbf{v}$ denotes the convection velocity, and f is the source term. The direct or forward problem for the convection-diffusion equation is well understood. The concentration distribution of the physical quantity (i.e. u) in space and time can be computed numerically by methods such as the finite difference method or the finite element method. However, the corresponding inverse problems, which include the parameter identification, source identification, boundary control and backward inverse problem, etc., are usually more difficult and expensive to solve than the forward problem for their ill-posedness in the sense of Hadamard [1,2]. The lack of stability with respect to the measurement data is a major issue, which means that small noises in the data may lead to significant changes in the reconstructed target. We shall focus on the inverse problem of space-time dependent source identification problem by using some given noisy measurement data of u. Several works have been devoted to the existence, uniqueness and stability estimates of a solution to this parabolic-type inverse problem. Mathematical properties of reconstructing a space-dependent source in the parabolic equation are found in [3] using a final moment measurement. Stability estimates are provided in [4] for recovering a space-time variable separable source term. Convergence analysis of the finite element approximation to the solution of an inverse source problem for parabolic equations is demonstrated in [5] with boundary observations and in [6] with inner integral observations respectively. In [7] the existence of the optimal solution to the regularized system for simultaneously reconstructing both the convection velocity and source strength in a convection-diffusion equation is proved and rigorous convergence analysis of the finite element method is established. Overall speaking, the mathematical analysis is problem dependent, regarding what observation data or overdetermined condition given for the inverse problem and depending on the

[*] Corresponding author.
E-mail addresses: xmdeng@oamail.gdufs.edu.cn (X. Deng), liaozj@jnu.edu.cn (Z.-J. Liao), xccai@um.edu.mo (X.-C. Cai).

https://doi.org/10.1016/j.camwa.2022.02.010
Received 25 September 2021; Received in revised form 22 January 2022; Accepted 18 February 2022
Available online 3 March 2022

specific form of the source function. For surveys on the subject, we refer the reader to the papers [3–10] and the references therein.

Source identification problems govern by the convection-diffusion equation arise in many applications including environmental problems, such as the detection and monitoring of indoor [11] and outdoor [12] air pollution, pollutant source tracking in underground water [13] or surface water [14], etc. The algorithms for inverse source problems have been investigated extensively [7,15–19]. The degree of ill-posedness of the inverse source problem for the convection-diffusion equation is investigated numerically using singular value decomposition of the input-output operators in [20]. The inverse problem is usually reformulated into a least-square optimization problem complemented with a regularization term, and classical optimization methods such as regression methods [21], linear and nonlinear programming methods [21], linear and nonlinear conjugate gradient methods [17,22,23], Newton type methods [24], can be used to obtain the approximate solutions.

Recent developments for inverse source identification problems can be summarized in three areas. The first one focuses on more sophisticated mathematical models, such as the time-fractional diffusion equation, which has gained increasing interests for its applications to describe the anomalous diffusion phenomena in highly heterogeneous media [25–27]. For instance, the space component of the source term for a time-fractional convection-diffusion equation was reconstructed by an iterative threshold algorithm in [25]. Secondly, the machine learning methods are increasingly applied as well as the traditional numerical methods in scientific computing. For example, a least-squares support-vector machines regression method is used for recovering the separable variable source function, where the solution of the 1D convection-diffusion system and the unknown source term are expressed using a kernel function over the training set [28]. A parallel genetic algorithm is employed to initialize the Nelder-Mead simplex method to estimate the source term of the atmospheric release [29]. Lastly, parallel algorithms with high efficiency and accuracy are in great need by large-scale applications, in pace with the rapid development of high performance supercomputers. Since the inverse source identification problem is usually reformulated as an optimization problem constrained with PDEs, sequential quadratic programming (SQP) methods are widely used, where the state variable, the adjoint variable and the source variable are iteratively updated in a sequential order [15]. SQP methods require low memory, but the number of iterations for a specified accuracy grows quickly with the increase of the optimization variables. Recently, parallel methods for solving the fully coupled Karush-Kuhn-Tucker (KKT) system become popular for its high efficiency in obtaining all the unknown variables all-at-once. For recovering space-time dependent sources, a space-time parallel Krylov-Schwarz method is developed in [30,31], in which the spatial and temporal variables are treated equally in a four-dimensional domain. The fully coupled KKT system is solved with a Krylov subspace method preconditioned by an additive Schwarz preconditioner in parallel. The space-time domain decomposition method offers higher degree of parallelism, shortens the total computing time at the expense of more memory.

In this paper, we develop a two-level domain decomposition method to further improve the efficiency of the parallel space-time domain decomposition method. Based on the idea of "divide and conquer", we firstly divide the original inverse problem for tracking the global source into smaller subproblems by partitioning the computational domain into several overlapping subdomains, and the sources defined on these subdomains are reconstructed by solving these subproblems independently. As will show in the paper, by introducing the domain decomposition, the space-time coupled large-scale KKT system is decoupled into several smaller-scale KKT systems. These subsystems can be solved simultaneously by a parallel space-time Krylov-Schwarz method, where a second level of domain decomposition on each subdomain is performed when applying the Schwarz preconditioner. Finally the solutions of all subsystems are appropriately glued together to form a global solution of the inverse source problem.

The rest of the paper is arranged as follows: Section 2 describes the mathematical formulation of the inverse problem. Section 3 proposes the two-level domain decomposition method. Three numerical examples are shown in Section 4 to test the reconstruction effect and the algorithm efficiency. Some conclusions are drawn in Section 5.

2. Mathematical formulation of the inverse source problem

We consider the following convection-diffusion equation in a open bounded and simply connected domain $\Omega \in \mathbb{R}^3$:

$$\begin{cases} \dfrac{\partial u}{\partial t} - \nabla \cdot (a\nabla u(\mathbf{x},t)) + \nabla \cdot (\mathbf{v}u(\mathbf{x},t)) = f(\mathbf{x},t), \qquad \mathbf{x}\in\Omega,\ t\in(0,T], \\ u(\mathbf{x},t) = p(\mathbf{x}), \qquad \mathbf{x}\in\Gamma_1, \\ \dfrac{\partial u}{\partial \mathbf{n}} = 0, \qquad \mathbf{x}\in\Gamma_2, \\ u(\mathbf{x},0) = u_0(\mathbf{x}), \end{cases} \tag{1}$$

where the Dirichlet boundary condition and homogeneous Neumann boundary condition are imposed on Γ_1 and Γ_2 respectively, with $\Gamma_1 \cup \Gamma_2 = \partial\Omega$. The initial condition is denoted as $u_0(\mathbf{x})$. The general source function $f(\mathbf{x},t)$ denotes the intensity of the moving source at each moment, which is the target to be recovered from the noisy measurement data of the concentration $u(\mathbf{x},t)$.

Defining the space-time domain $\Omega_T = \Omega \times [0,T]$, suppose $f(\mathbf{x},t) \in L^2(\Omega_T)$, $p(\mathbf{x}) \in L^2(\Gamma_1)$ and $u_0(\mathbf{x}) \in L^2(\Omega)$, then from the standard parabolic theory [32], there exists a unique weak solution to (1). Denoting $(\cdot,\cdot)$ as the L^2 inner product, then u satisfies the following variational equation:

$$\left(\frac{\partial u}{\partial t}, \phi\right) + (a\nabla u, \nabla\phi) - (\mathbf{v}u, \nabla\phi) = (f, \phi), \tag{2}$$

$\forall \phi \in L^2(0,T; H^1_{\Gamma_1}(\Omega))$, with $H^1_{\Gamma_1}(\Omega)$ defined by

$$H^1_{\Gamma_1}(\Omega) = \{\varphi | \varphi \in H^1(\Omega), \text{ the trace of } \varphi \text{ vanishes on } \Gamma_1.\}$$

Furthermore u has the following regularity:

$$u \in L^2(0,T; H^1(\Omega)) \cap H^1(0,T; L^2(\Omega)). \tag{3}$$

We are interested in how the solution of the parabolic equation is affected by the variation of the Dirichlet boundary condition. Suppose u_1 satisfies the same equation (1) with only a different Dirichlet boundary condition $p_1(\mathbf{x})$ given on Γ_1. Defining $\tilde{u} = u_1 - u$ and $\tilde{p}(\mathbf{x}) = p_1(\mathbf{x}) - p(\mathbf{x})$, then $\tilde{u}$ satisfies the following equation:

$$\begin{cases} \dfrac{\partial \tilde{u}}{\partial t} - \nabla \cdot (a\nabla \tilde{u}(\mathbf{x},t)) + \nabla \cdot (\mathbf{v}\tilde{u}(\mathbf{x},t)) = 0, \qquad \mathbf{x}\in\Omega, t\in(0,T], \\ \tilde{u}(\mathbf{x},t) = \tilde{p}(\mathbf{x}), \qquad \mathbf{x}\in\Gamma_1, \\ \dfrac{\partial \tilde{u}}{\partial \mathbf{n}} = 0, \qquad \mathbf{x}\in\Gamma_2, \\ \tilde{u}(\mathbf{x},0) = 0. \end{cases} \tag{4}$$

By performing the energy estimate [32], there exists a constant C such that:

$$\|\tilde{u}\|_{L^2(0,T;H^1(\Omega))} \le C\|\tilde{p}\|_{L^2(\Gamma_1)}, \tag{5}$$

i.e.,

$$\|u_1 - u\|_{L^2(0,T;H^1(\Omega))} \le C\|p_1 - p\|_{L^2(\Gamma_1)}. \tag{6}$$

From (6) we see that if there is a sequence $p_n \to p$ strongly in $L^2(\Gamma_1)$, the corresponding weak solution $u_n \to u$ strongly in $L^2(0,T;H^1(\Omega))$.

2.1. The Tikhonov regularization method

Denoting the measurement data by $u^{\epsilon}(\mathbf{x},t)$, with ϵ being the noise level, the inverse source problem is reformulated into an output least-square optimization problem. The objective functional, consisting of a data-fitting term and a regularization term to ensure the well-posedness of the optimization problem [33], can be described as:

$$J(f)=\frac{1}{2}\int_0^T\int_\Omega A(\mathbf{x})(u(\mathbf{x},t)-u^{\epsilon}(\mathbf{x},t))^2\,d\mathbf{x}dt+N_{\beta}(f), \tag{7}$$

where $A(\mathbf{x})$ is the data range indicator function to specify the measurement locations. $N_{\beta}(f)$ in (7) is chosen as the following H^1-H^1 Tikhonov regularization in both space and time:

$$N_{\beta}(f)=\frac{\beta_1}{2}\int_0^T\int_\Omega |\dot{f}|d\mathbf{x}dt+\frac{\beta_2}{2}\int_0^T\int_\Omega |\nabla_{\mathbf{x}}f|^2d\mathbf{x}dt, \tag{8}$$

where β_1 and β_2 are two regularization parameters. Other regularizations, such as H^1-L^2, L^2-L^2 may be used, depending on the availability of prior information of the source function. Now the inverse source problem is equivalent to solving the following constrained optimization problem $\mathcal{P}$:

$$\mathcal{P}:\quad \min_{f\in L^2(\Omega_T)} J(f),\qquad \text{subject to } (u,f) \text{ satisfying (1) in } \Omega_T. \tag{9}$$

By introducing a Lagrange multiplier or an adjoint variable v, the optimization problem (9) is transformed into an unconstrained minimization problem with the objective functional revised as:

$$\mathcal{J}(u,f,v)=J(f)+(v,L(u,f)), \tag{10}$$

where $L(u,f)$ denotes the convection-diffusion operator in (1) and $(v,L(u,f))$ stands for their L^2 inner product. The necessary condition satisfied by the minima of (10) is the first-order optimality conditions, the so-called KKT system. The weak formulation of the KKT system is obtained by taking variations with respect to v, u and f and computing the corresponding Fréchet derivatives as

$$\begin{cases}\mathcal{J}_v(u,f,v)\phi=0,\\ \mathcal{J}_u(u,f,v)\psi=0,\\ \mathcal{J}_f(u,f,v)\zeta=0,\end{cases} \tag{11}$$

for all test functions ϕ, ψ and ζ. Then using integration-by-part, we obtain the following weak-form of the KKT system:

$$\begin{cases}\left(\dfrac{\partial u}{\partial t},\phi\right)+(a\nabla u,\nabla\phi)-(\mathbf{v}u,\nabla\phi)=(f,\phi),\\ -\left(\dfrac{\partial v}{\partial t},\psi\right)+(a\nabla v,\nabla\psi)+(\nabla\cdot(\mathbf{v}\psi),v)=(A(\mathbf{x})(u^{\epsilon}-u),\psi),\\ -(v,\zeta)+\beta_1\left(\dot{f},\dot{\zeta}\right)+\beta_2(\nabla f,\nabla\zeta)=0,\end{cases} \tag{12}$$

$\forall\phi,\psi\in L^2(0,T;H^1_{\Gamma_1}(\Omega))$, and $\zeta\in H^1(0,T;H^1(\Omega))$. The three equations of (12) are called the state, the adjoint and the source equation respectively. As we observe, the state and the adjoint equations are standard parabolic PDEs, we use a second-order Crank-Nicolson finite difference scheme in time and a piecewise linear finite element method in space to discretize both equations. Since the source equation is elliptic in the space-time coupled domain Ω_T, a 4D piecewise linear finite element in both space (three dimensions) and time (one dimension) is applied with a uniform partition in Ω_T [30]. Consider a regular triangulation $\mathcal{T}^h$ of Ω and a uniform time partition $\mathcal{P}^{\tau}$ in $[0,T]$: $0=t^0<t^1<\cdots<t^M=T$, with $t^k=k\tau$, $\tau=T/M$. $\mathcal{T}^h\times\mathcal{P}^{\tau}$ is applied to generate the partition in the space-time domain $\Omega_T=\Omega\times[0,T]$. Then we introduce the piecewise linear finite element space V^h on $\mathcal{T}^h$ and W_h^{τ} on $\mathcal{T}^h\times\mathcal{P}^{\tau}$. Denoting the difference quotient and the averaging of a function $g(\mathbf{x},t)$ as:

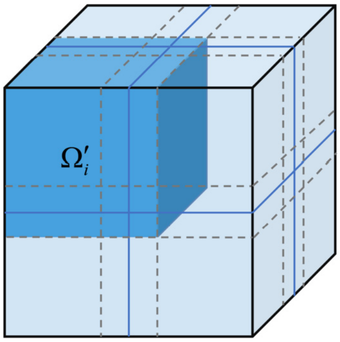

Fig. 1. The overlapping domain decomposition of a cubic domain Ω into Ω_i', $i=1,\cdots,nd$.

$$\partial_{\tau}g^k(\mathbf{x})=\frac{g^k(\mathbf{x})-g^{k-1}(\mathbf{x})}{\tau},\qquad \bar{g}^k(\mathbf{x})=\frac{1}{\tau}\int_{t^{k-1}}^{t^k}g(\mathbf{x},t)dt,$$

the discrete finite element system of (12) can be formulated as: Find the sequence of approximations u_h^k, $v_h^k\in V^h$ and $f_h^k\in W_h^{\tau}$ such that

$$\begin{cases}\left(\partial_{\tau}u_h^k,\phi_h\right)+\left(a\nabla\bar{u}_h^k,\nabla\phi_h\right)-\left(\mathbf{v}\bar{u}_h^k,\nabla\phi_h\right)=\left(\bar{f}_h^k,\phi_h\right),\\ -\left(\partial_{\tau}v_h^k,\psi_h\right)+\left(a\nabla\bar{v}_h^k,\nabla\psi_h\right)+\left(\nabla\cdot\left(\mathbf{v}\psi_h\right),\bar{v}_h^k\right)=\left(A(\mathbf{x})\left(\bar{u}_h^{\epsilon,k}-\bar{u}_h^k\right),\psi_h\right),\\ -\left(v_h^k,\zeta_h^k\right)+\beta_1\left(\partial_{\tau}f_h^k,\partial_{\tau}\zeta_h^k\right)+\beta_2\left(\nabla f_h^k,\nabla\zeta_h^k\right)=0,\end{cases} \tag{13}$$

$\forall\phi_h,\psi_h\in\mathring{V}^h$, and $\zeta_h^k\in W_h^{\tau}$, where $\mathring{V}^h$ denotes the subspace of V^h with zero trace on Γ_1. We remark that the computational cost of solving the fully-coupled KKT system (13) can be tremendous.

3. The two-level domain decomposition method

To avoid solving such a large linear system (13), we propose a two-level domain decomposition method for solving the optimization problem (9). Firstly, we decompose the optimization problem into smaller-scale subproblems by dividing the domain Ω into nd non-overlapping subdomains Ω_1, Ω_2, $\cdots$, Ω_{nd}, see Fig. 1 for a demonstration of the domain decomposition of a cubic domain, with the black lines and the blue lines to differentiate the actual boundary of Ω and the artificial boundary of the subdomains. From the additivity of the integral, the objective functional (7) can be reformulated as

$$J(f)=\sum_{i=1}^{nd}\frac{1}{2}\int_0^T\int_{\Omega_i}A(\mathbf{x})(u(\mathbf{x},t)-u^{\epsilon}(\mathbf{x}),t)^2\,d\mathbf{x}dt+N_{\beta}(f_i), \tag{14}$$

with f_i being the source f restricted in Ω_i. Defining the objective functional $J(f_i)$ on Ω_i as

$$J(f_i)=\frac{1}{2}\int_0^T\int_{\Omega_i}A(\mathbf{x})(u(\mathbf{x},t)-u^{\epsilon}(\mathbf{x},t))^2\,d\mathbf{x}dt+N_{\beta}(f_i), \tag{15}$$

then $J(f)=\sum_{i=1}^{nd}J(f_i)$. As a result, instead of solving (9), we focus on solving the following subproblems $\mathcal{P}_i$, $i=1,\cdots,nd$:

$$\mathcal{P}_i:\quad \min_{f_i\in L^2(\Omega_i\times[0,T])} J(f_i),\qquad \text{subject to } (u,f_i) \text{ satisfying (1) in } \Omega_i\times[0,T]. \tag{16}$$

For the subproblem (16), appropriate boundary conditions are needed on the artificial boundaries. It is observed from (6) that changes in the solution of the convection-diffusion equation are bounded by the variety of the Dirichlet boundary condition. If simple homogeneous Dirichlet boundary conditions are imposed, the solution around these

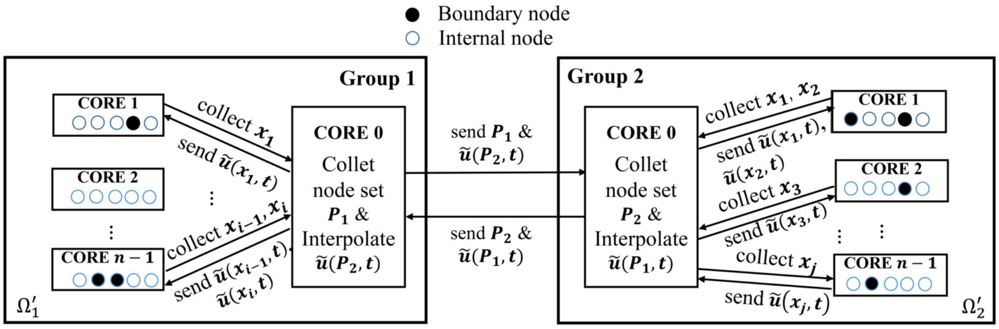

Fig. 2. The interpolation and communication process between neighboring subdomains for correcting the artificial boundary condition.

artificial boundaries can be quite different from the true solution, which can be regarded as a forward model error.

To overcome this difficulty, we revise the domain decomposition into an overlapping one $\{\Omega_i'\}$, $i=1,\cdots,nd$. As shown in Fig. 1, the subdomain edge is marked as dotted gray lines. On each subdomain Ω_i', similarly by using the Lagrange multiplier method, we obtain the KKT system for $\mathbf{x}\in\Omega_i'$, and $t\in(0,T)$:

$$\begin{cases}\left(\dfrac{\partial u}{\partial t},\phi\right)+(a\nabla u,\nabla\phi)-(\mathbf{v}u,\nabla\phi)=\left(f_i,\phi\right),\\ -\left(\dfrac{\partial v}{\partial t},\psi\right)+(a\nabla v,\nabla\psi)+(\nabla\cdot(\mathbf{v}\psi),v)=(A(\mathbf{x})(u^\epsilon-u),\psi),\\ -(v,\zeta)+\beta_1\left(\dot{f}_i,\dot{\zeta}\right)+\beta_2(\nabla f_i,\nabla\zeta)=0,\end{cases}\tag{17}$$

$\forall\phi,\psi\in L^2(0,T;H^1_{\tilde{\Gamma}_i}(\Omega_i'))$ and $\zeta\in H^1(0,T;H^1(\Omega_i'))$. The boundary of each subdomain Ω_i', denoted by $\partial\Omega_i'$ may consist of three parts: the actual boundary shared with Γ_1 and Γ_2, as well as the artificial boundary generated by the decomposition, denoted by Γ_i'. Here we denote $\tilde{\Gamma}_i=(\partial\Omega_i'\cap\Gamma_1)\cup\Gamma_i'$ as the Dirichlet part of $\partial\Omega_i'$. The boundary, initial and terminal conditions for u, v and f_i are determined by two steps.

Firstly, the state variable, i.e., the concentration u maintain the same boundary conditions on $\partial\Omega_i'\backslash\Gamma_i'$ and the same initial condition given by (1) as

$$\begin{cases}u(\mathbf{x},t)=p(\mathbf{x}), & \mathbf{x}\in\left(\partial\Omega_i'\cap\Gamma_1\right),\\ \dfrac{\partial u}{\partial\mathbf{n}}=0, & \mathbf{x}\in\left(\partial\Omega_i'\cap\Gamma_2\right),\\ u(\mathbf{x},0)=u_0(\mathbf{x}), & \mathbf{x}\in\Omega_i'.\end{cases}\tag{18}$$

And by the arbitrariness of ψ, the boundary conditions on $\partial\Omega_i'\backslash\Gamma_i'$ and the terminal condition for v are derived as:

$$\begin{cases}v(\mathbf{x},t)=0, & \mathbf{x}\in\left(\partial\Omega_i'\cap\Gamma_1\right),\\ \dfrac{\partial v}{\partial\mathbf{n}}+(\mathbf{v}\cdot\mathbf{n})v=0, & \mathbf{x}\in\left(\partial\Omega_i'\cap\Gamma_2\right),\\ v(\mathbf{x},T)=0, & \mathbf{x}\in\Omega_i'.\end{cases}\tag{19}$$

Similarly using the arbitrariness of ζ, we obtain the boundary conditions on $\partial\Omega_i'\backslash\Gamma_i'$, the initial and terminal conditions for f_i as:

$$\begin{cases}\dfrac{\partial f_i}{\partial\mathbf{n}}=0 & \text{for }\ \mathbf{x}\in\left(\partial\Omega_i'\cap\partial\Omega\right),\\ \dfrac{\partial f_i}{\partial t}=0 & \text{for }\ t=0,T,\ \mathbf{x}\in\Omega_i'.\end{cases}\tag{20}$$

Secondly, on the artificial boundary Γ_i', we propose two kinds of boundary conditions: one is to give the homogeneous boundary conditions for u, v and f_i manually as follows:

$$\begin{cases}u(\mathbf{x},t)=0, & \mathbf{x}\in\Gamma_i',\\ v(\mathbf{x},t)=0, & \mathbf{x}\in\Gamma_i',\\ \dfrac{\partial f_i}{\partial\mathbf{n}}=0, & \mathbf{x}\in\Gamma_i'.\end{cases}\tag{21}$$

Since the realistic values of u on the artificial boundary are usually not zero, (21) gives only a rough approximation. However, we know from (6) that the difference of the solutions with different Dirichlet boundary conditions for the parabolic equation is in the same magnitude of the difference of the Dirichlet boundary conditions, it can be expected that the reconstruction error of the inverse problem would decrease as the increase of the overlapping size among the subdomains [34]. On the other hand, a more accurate setting of the artificial Dirichlet boundary is using solutions of the relevant neighboring subdomains from the last iteration, as in [35]. Considering the communication cost in parallel computing, instead of updating the boundary values after each iterative step of the linear solver, in our algorithm we use an estimation-correction approach for obtaining the values of $u(\mathbf{x},t)$ on Γ_i'. On the first stage, homogeneous boundary conditions as in (21) are imposed, and by solving the KKT system (17) an estimation of the boundary value $\tilde{u}(\mathbf{x}_l,t)$ on Γ_i' is obtained by interpolation from some neighboring subdomain Ω_j' where $\mathbf{x}_l\in\Omega_j$. It is noted that each $\mathbf{x}_l$ belongs to only one adjacent subdomain Ω_j. Then the revised value $\tilde{u}(\mathbf{x}_l,t)$ is sent from Ω_j' to Ω_i' to correct the corresponding boundary value on Γ_i'. As long as all the boundary values on Γ_i' are received and revised by their corresponding neighboring subdomains, the KKT system (17) is solved once again with the corrected boundary conditions to obtain an improved solution for the unknown source function. Taking two neighboring subdomains for example in Fig. 2, two groups of n processor cores ("Group 1" and "Group 2") are in charge of computing the subproblems on Ω_1' and Ω_2' respectively. Inside Group 1, the processor cores are sorted and the first processor core ("CORE 0") takes the responsibility of collecting the coordinates of all inner boundary nodes (black circles) on Ω_1' and the node sets is denoted by "$\mathbf{P_1}$". The same operations are performed in Group 2 and the inner boundary node set "$\mathbf{P_2}$" is collected. Then "CORE 0" of Group 1 and "CORE 0" of Group 2 exchange their node sets by MPI communications. For those nodes of $\mathbf{P_1}$ in the non-overlapping region of Ω_2', "CORE 0" of Group 2 interpolates from the solution values of its own group to obtain the updated boundary values $\tilde{u}(\mathbf{P_1},t)$, and does nothing for those nodes in the overlapping region of Ω_2'. The same operations are performed in Group 1 to get the updated boundary values $\tilde{u}(\mathbf{P_2},t)$. Two-way communications are established again to exchange $\tilde{u}(\mathbf{P_1},t)$ and $\tilde{u}(\mathbf{P_2},t)$. Finally "CORE 0" of both groups send all the updated values to the other processor cores in their own Group (denoted by "CORE 1", "CORE 2", $\cdots$, "CORE $n-1$") to correct the previously stored boundary values.

The subproblems (17) on Ω_i', $i=1,\cdots,nd$ are discretized similarly as in (13) and solved in parallel by a space-time Krylov-Schwarz method. To form the additive Schwarz preconditioner, a second level of overlapping space-time domain decomposition is performed inside each subdomain $\Omega_i'\times[0,T]$, $i=1,2,\cdots,nd$. The left of Fig. 3 provides an example of the space decomposition inside Ω_i' and the right of Fig. 3 demonstrates the space-time overlapping decomposition. For the detail of the space-time domain decomposition method, we refer to [30]. By choosing appropriate overlapping size, we cut out all solutions corresponding to the overlapping region between neighboring subdomains, i.e. $\Omega_i'\backslash\Omega_i$, $i=1,\cdots,nd$, then the solutions in $\bigcup_{i=1}^{nd}\Omega_i$ consist an approximate solution to the original problem $\mathcal{P}$. The algorithm is described as follows:

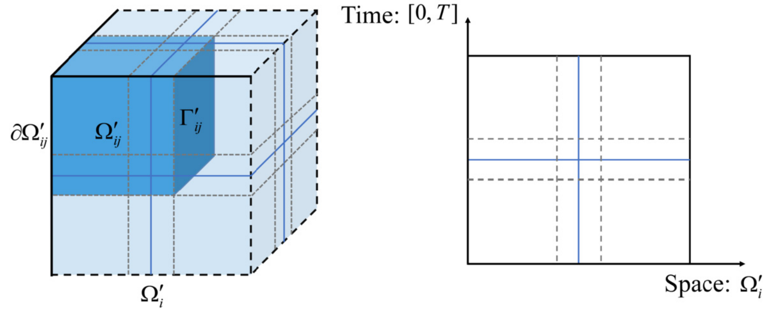

Fig. 3. Left: the overlapping domain decomposition of the subdomain Ω'_i $(i=1,\cdots,nd)$ into Ω'_{ij}, $(j=1,\cdots,n)$. Right: the space-time domain decomposition in the subdomain $\Omega'_i\times[0,T]$.

Algorithm 1 A two-level domain decomposition method.

```
1: Decompose $\Omega$ into overlapping subdomains $\Omega'_1, \Omega'_2, \cdots, \Omega'_{nd}$;
2: For each subdomain $\Omega'_i$, $i=1, \cdots, nd$ (in parallel):
    2.1) Decompose $\Omega'_i$ into overlapping subdomains $\Omega'_{i1}, \Omega'_{i2}, \cdots, \Omega'_{in}$;
    2.2) Apply the space-time Krylov-Schwarz method to solve the optimization problem
         (16) (in parallel) with homogeneous boundary conditions in (21) imposed on $\Gamma'_i$ or
         perform an estimation-correction process in Fig. 2 and obtain $f_i$;
3: Cut off the parts of $f_i$, $i=1,\cdots,nd$ in the region $\Omega'_i\backslash\Omega_i$ and put the non-overlapping
   parts together to obtain $f$.
```

4. Numerical experiments

In this section several numerical experiments are given to study the accuracy, robustness and parallel efficiency of the proposed algorithm. In our computations, the computational domain is taken to be $\Omega=[-L,\,L]\times[-S,\,S]\times[-H,H]$ with $L=S=H=2$, the terminal time is $T=1$ unless otherwise specified and the initial condition is $u_0(\mathbf{x})=0$. The boundary is decomposed into $\Gamma_1:\{\mathbf{x}=(x_1,x_2,x_3)||x_1|=L \text{ or } |x_2|=S\}$ and $\Gamma_2:\{\mathbf{x}=(x_1,x_2,x_3)||x_3|=H\}$. The Dirichlet boundary condition on Γ_1 is $p(\mathbf{x})=0$. The diffusion and convection coefficients are set to be $a=1.0$ and $\mathbf{v}=(1.0,1.0,1.0)^T$.

In order to generate the observation data, we solve the convection-diffusion equation (1) on a very fine mesh to obtain the numerical solution $u_h^\tau(\mathbf{x},t)$. The data range indicator function $A(\mathbf{x})$ is chosen as:

$$A(\mathbf{x})=\sum_{i=1}^{nd}\sum_{j=1}^{N_i}\delta(\mathbf{x}-\mathbf{x}_{ij}),$$

where $\mathbf{x}_{ij}$, $j=1,\cdots,N_i$ specify the measurement locations on a uniform sparse mesh of each subdomain Ω'_i, and $\delta(\mathbf{x}-\mathbf{x}_{ij})$ is the Dirac delta function. The values of $u(\mathbf{x}_{ij},t)$ are obtained by applying a linear interpolation operator $\mathcal{I}$ to the simulated solution $u_h^\tau(\mathbf{x},t)$. Then the data noise is generated by adding Gaussian noises to the resulting approximate solution $u(\mathbf{x}_{ij},t)$ as follows:

$$u^\epsilon(\mathbf{x}_{ij},t)=u(\mathbf{x}_{ij},t)+\epsilon\,\sigma\,u(\mathbf{x}_{ij},t),\quad i=1,\cdots,nd,\ j=1,\cdots,N_i,$$

where σ is a random function satisfying the standard normal distribution and ϵ is the noise level, $\epsilon=1\%$ if not otherwise specified.

We test the following three examples of source functions:

Example 1.

$$f(x_1,x_2,x_3,t)=\sin\left(\frac{\pi x_1x_2x_3t}{LSH}\right)\cos\left(\frac{\pi x_1x_2x_3t}{LSH}\right)+1.0,$$

Example 2.

$$f(x_1,x_2,x_3,t)=(L-(1.0-t)x_1)+(S-x_2t)+(H-x_3)t^2,$$

Example 3.

$$f(x_1,x_2,x_3,t)=\begin{cases}3.0, & \text{if}\quad \left(\frac{x_1+0.5L\sin(1.5\pi t)}{0.4L}\right)^2+\left(\frac{x_2+0.5S\cos(1.5\pi t)}{0.4S}\right)^2\\ & \qquad +\left(\frac{x_3-0.5H(1-2t)}{0.2H}\right)^2\le 1,\\ 1.0, & \text{otherwise.}\end{cases}$$

For the above three examples, we show the reconstructed source profiles and test their solution accuracy with respect to different noise levels, terminal time and iteration stopping criterion. Furthermore we compare the results with the two kinds of artificial boundary conditions by modifying the amount of measurement data and the overlapping size between neighboring subdomains. Finally, the parallel efficiency is indicated by scalability test with different number of subdomains. The numerical experiments are implemented by PETSc (the Portable, Extensible Toolkit for Scientific Computation) [36]. All programs are run on “Tianhe-2A” which is a 61.44-petaflops supercomputer located in the National Supercomputer Center in Guangzhou, China.

4.1. Numerical reconstruction with noise level test

We divide the cubic domain uniformly into $nd=8$ subdomains, each subdomain is of length 2.0, width 2.0 and height 2.0. Then the subdomains are revised into overlapping ones with an overlap ratio γ that defined by the overlapping size over the subdomain edge width. In this numerical experiment we set $\gamma=0.5$, i.e., the overlapping sizes between neighboring subdomains are equally 1.0 in x_1, x_2 and x_3 dimension. The mesh used for inversion on each subdomain is $33\times33\times33$, and the number of time step is $M=33$. Homogeneous boundary conditions are imposed on the artificial boundaries. 27 processor cores are used to solve each subproblem and the total number of processor cores is $np=216$. The ratio of choosing the measurement data is $r=1/2$, i.e., $16\times16\times16$ at each subdomain. The discrete KKT systems are solved by a restarted GMRES method (restart at 50). The stopping criterion for GMRES iteration is the relative convergence tolerance $rtol$, i.e., the decrease of the residual norm relative to the norm of the right hand side (in l^2 norm), and the tolerance is set as $rtol=10^{-6}$ if not otherwise specified. When applying the space-time additive Schwarz preconditioner, a second level of overlapping domain decomposition is performed inside each subdomain. The overlapping sizes are set as $2h$ in space dimensions and 2τ in time dimension, where h and τ are denoted as the mesh size and the time step size respectively. The initial guesses for u, v and f_i $(i=1,\cdots,nd)$ are uniformly zero. The true solution and the numerical solution are shown in Fig. 4 - 6. To better compare the numerical solution and the exact solution in the space-time 4D domain, these figures are presented in three different ways. We show the distribution of the source function at three moments $t=0$, 0.5 and 1.0 and at three slices $x=-2$, 0 and 2 in Fig. 4 for Example 1. It is observed that the numerical results at $t=0.5$ and $t=1.0$ are quite accurate, while the numerical solution for the initial moment $t=0$ is less satisfactory, the

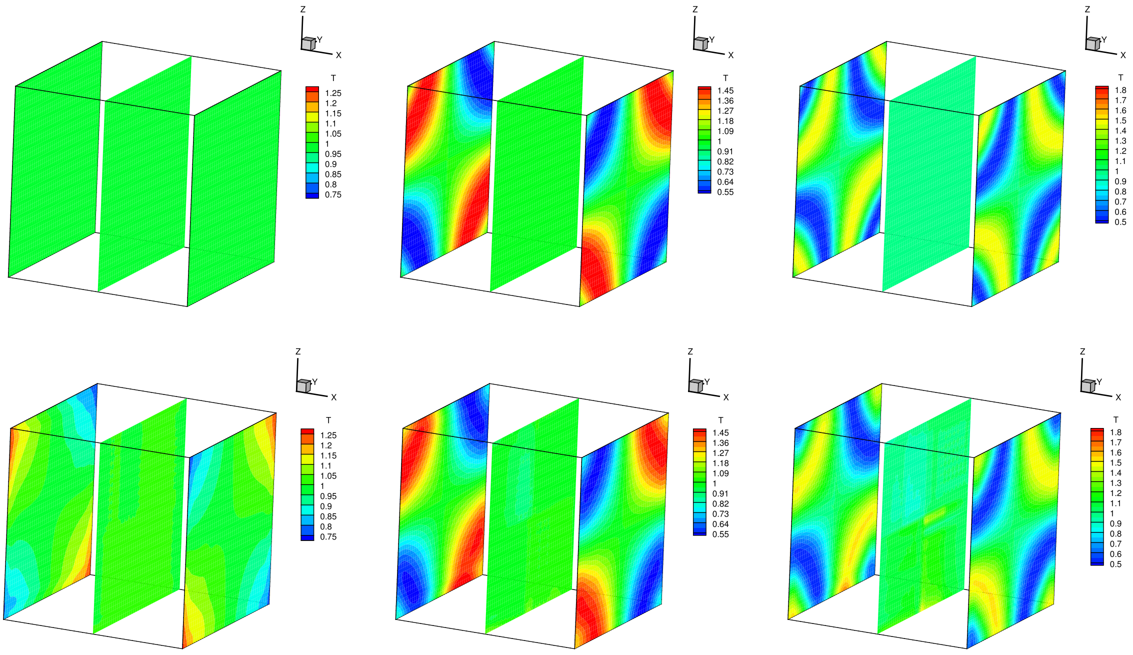

Fig. 4. Example 1: Comparison of the slices of the exact source profile (top) at $x = -2, 0, 2$ and the numerical ones (bottom) at the moments $t = 0$ (left), 0.5 (middle) and 1.0 (right).

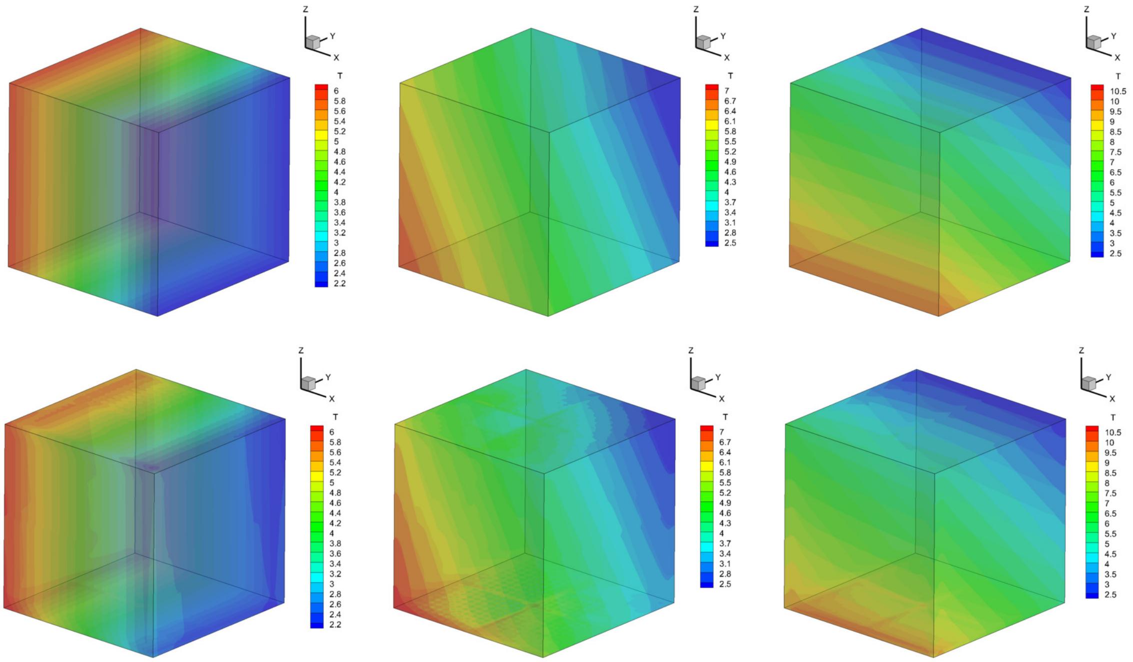

Fig. 5. Example 2: Comparison of the exact source profile (top) at the domain surfaces and the numerical ones (bottom) at the moments $t = 0$ (left), 0.5 (middle) and 1.0 (right).

corners of the slices $x = -2$ and 2 are contaminated by accumulated numerical errors. For Example 2, we plot the numerical results on the six domain surfaces at three moments $t = 0$, 0.5 and 1.0 in Fig. 5. It is shown that both the distribution and intensity of the source are accurately reconstructed. Since Example 3 is a piecewise constant ellipsoid source, the isosurfaces at three moments $t = 0$, 0.5 and 1.0 are plotted in Fig. 6. Both the location and the scale of the source are well found.

To show the numerical error, we define the following error function

$$E = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\left(f(\mathbf{x}_i,t_j) - f^*(\mathbf{x}_i,t_j)\right)^2}{\sum_{i=1}^{N}\sum_{j=1}^{M}\left(f^*(\mathbf{x}_i,t_j)\right)^2}},$$

where N denotes the summation number of mesh nodes in the non-overlapping region of all subdomains and M is the number of time steps. Then we test the numerical errors with different noise levels $\epsilon = 0\%, 1\%, 3\%, 5\%$ respectively. The regularization parameters β_1 and β_2 in the computation for each case are selected heuristically. β_1 and β_2 are set with a same value β, and the best fit of the parameter is chosen from six candidates $\beta = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$. We show

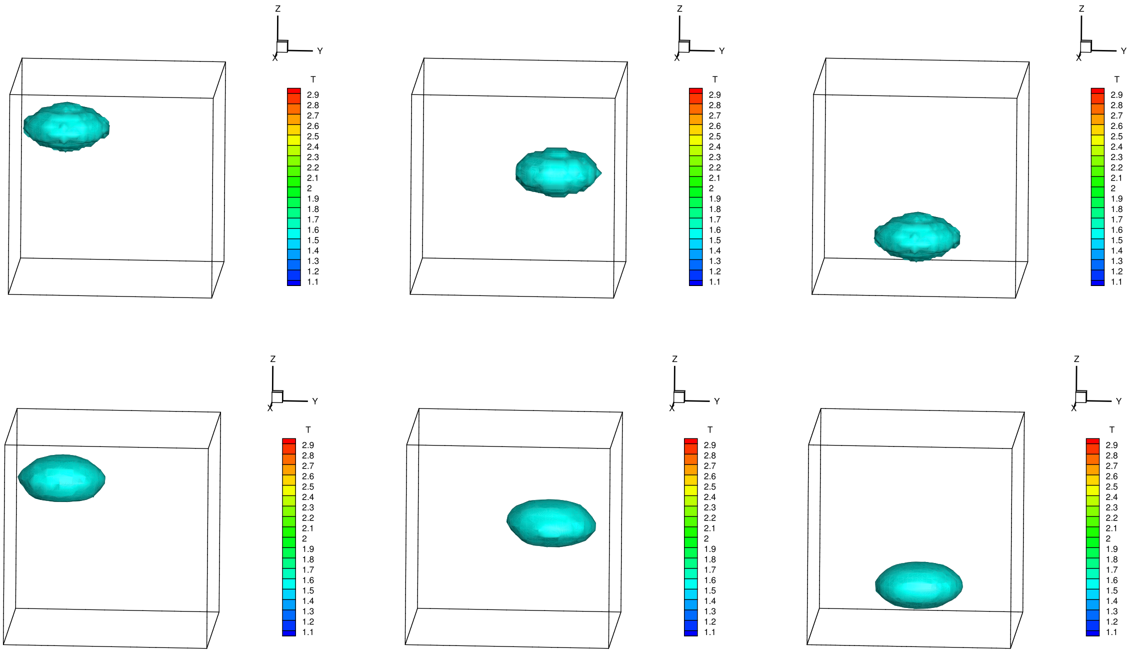

Fig. 6. Example 3: Comparison of the isosurfaces of the exact source profile (top) and the numerical ones (bottom) at the moments $t = 0$ (left), 0.5 (middle) and 1.0 (right).

Table 1
The computing performance with increasing noise levels for Example 1-3.

ϵ	Ex1				Ex2				Ex3			
	β	N_{its}	T_{GMRES}	E	β	N_{its}	T_{GMRES}	E	β	N_{its}	T_{GMRES}	E
0%	10^{-5}	46	13.97	0.021	10^{-5}	45	13.96	0.017	10^{-5}	46	13.98	0.085
1%	10^{-4}	73	19.69	0.065	10^{-5}	46	14.08	0.049	10^{-5}	46	14.04	0.105
3%	10^{-3}	119	29.38	0.081	10^{-3}	120	29.50	0.063	10^{-4}	73	19.72	0.125
5%	10^{-3}	120	29.47	0.104	10^{-3}	121	29.67	0.074	10^{-3}	121	29.67	0.157

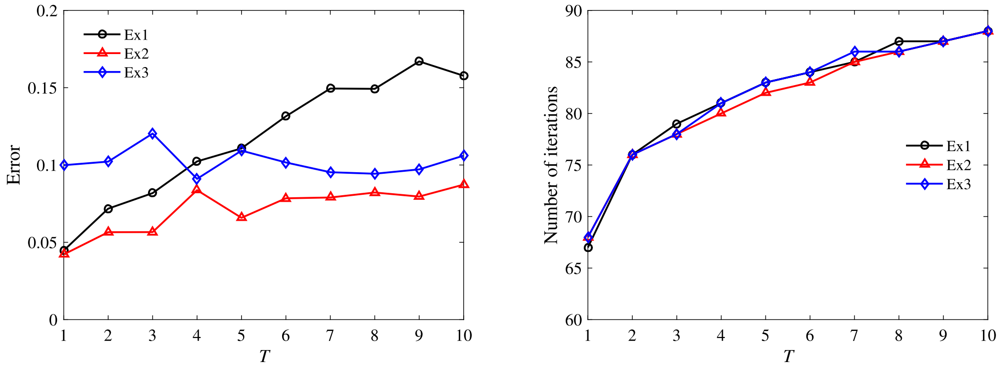

Fig. 7. The reconstruction errors (left) and the GMRES iteration numbers (right) with different terminal time.

the average number of GMRES iterations denoted by N_{its}, the average computing time denoted by T_{GMRES} (in sec), and the summation error E in Table 1. It is observed that the summation errors increase with the noise level, and the error of Example 3 is relatively larger than those of Example 1 and Example 2.

Next, we shall investigate how the reconstruction errors and the GMRES iterations vary with larger terminal time. The computational domain is also uniformly decomposed into $nd = 8$ subdomains. The space mesh for each subdomain is set as $37 \times 37 \times 37$. The noise level is $\epsilon = 1\%$ and according to the numerical tests, the regularization parameter is set as $\beta = 10^{-4}$ for all the three examples. The terminal time is extended from $T = 1$ to $2, \cdots, 10$ respectively and the time step size is set as $\tau = 0.05$ for all test cases. Correspondingly the number of time steps are $M = 21, 41, \cdots, 201$, and the number of processor cores used for the computation are $np = 216, 432, \cdots, 2160$ respectively. The reconstruction errors and the average number of GMRES iterations are drawn in Fig. 7. As shown in Fig. 7, the reconstruction error of the trigonometric function sources (Example 1) increases faster than the polynomial (Example 2) and the piecewise constant (Example 3) sources when the simulation duration gets larger, while the iteration number of GMRES method grows quite slowly with increasing terminal time for all three examples.

Table 2
The stopping criterion test with $nd = 8$ and $nd = 1$.

$rtol$	Ex1 ($nd = 8$)				Ex1 ($nd = 1$)			
	N_{its}	T_{GMRES}	T_{PRE}	E	N_{its}	T_{GMRES}	T_{PRE}	E
10^{-6}	67	14.69	9.82	0.044	89	19.82	46.87	0.057
10^{-5}	49	11.70	7.87	0.049	71	16.25	44.99	0.072
10^{-4}	35	9.17	6.49	0.041	45	11.57	42.05	0.064
10^{-3}	21	6.80	5.03	0.087	26	7.99	40.23	0.140
10^{-2}	11	5.27	4.19	0.672	11	5.43	38.67	1.413
$rtol$	Ex2 ($nd = 8$)				Ex2 ($nd = 1$)			
	N_{its}	T_{GMRES}	T_{PRE}	E	N_{its}	T_{GMRES}	T_{PRE}	E
10^{-6}	68	14.78	9.77	0.042	90	19.57	46.41	0.054
10^{-5}	50	11.80	8.16	0.040	70	15.94	44.28	0.041
10^{-4}	35	9.17	6.43	0.054	45	11.48	41.73	0.044
10^{-3}	22	7.00	5.19	0.078	25	7.96	39.86	0.117
10^{-2}	11	5.29	4.14	0.626	11	5.39	38.60	1.408
$rtol$	Ex3 ($nd = 8$)				Ex3 ($nd = 1$)			
	N_{its}	T_{GMRES}	T_{PRE}	E	N_{its}	T_{GMRES}	T_{PRE}	E
10^{-6}	68	14.75	9.75	0.099	91	19.43	45.76	0.113
10^{-5}	49	11.67	7.81	0.103	70	15.62	44.11	0.116
10^{-4}	35	9.11	6.38	0.103	47	11.86	41.17	0.112
10^{-3}	22	6.93	5.13	0.138	26	8.00	39.52	0.159
10^{-2}	11	5.27	4.17	0.698	11	5.43	37.68	1.459

4.2. Comparison of computing performance with $nd = 1$ and $nd = 8$

In this section we perturb the stopping criterion $rtol$ of GMRES method to compare the computing performance of the proposed algorithm with $nd = 8$ and the classical algorithm without domain decomposition at the first level, i.e. $nd = 1$. The time step M is set as 21 and the total number of processor cores is $np = 216$ for both algorithms. The space mesh of each subdomain for $nd = 8$ is set as $37 \times 37 \times 37$; to keep the same problem size, the space mesh for $nd = 1$ is set as $73 \times 73 \times 73$. Other parameters remain the same as the large terminal time test. Then we increase the stopping criterion $rtol$ from 10^{-6} to 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, and compare the number of GMRES iterations, the solving time T_{GMRES} (in sec), the preconditioning time T_{PRE} (in sec) and the reconstruction error E in Table 2. As observed from Table 2, the reconstruction errors are satisfying for $rtol = 10^{-6}$, 10^{-5} and 10^{-4}, then deteriorate for $rtol = 10^{-3}$, and become unreasonable for $rtol = 10^{-2}$. Furthermore, the reconstruction errors are compatible for the cases of $nd = 8$ and $nd = 1$; the errors for $nd = 1$ seem much more sensitive to $rtol$ than the case of $nd = 8$. Regarding to the solving time and the preconditioning time, they both reduce accordingly with increasing $rtol$ as expected. For the case of $nd = 1$, we find that the preconditioning time is dominant as it is several times of the solving time; for the case of $nd = 8$, the ratio of the time spending on the preconditioning stage versus the solving stage reduces significantly, and as a result both T_{GMRES} and T_{PRE} are much shorter than the case of $nd = 1$ (for $rtol < 10^{-2}$). Thus an obvious advantage of the proposed algorithm lies in that it can effectively relieve the burden of forming the global preconditioner, and the computing effort is much reduced when constructing eight subdomain preconditioners.

4.3. Comparison with two artificial boundary conditions

Next we compare the results of using the two different artificial boundary conditions, namely the homogeneous boundary condition and the estimation-correction boundary condition. The domain decomposition of Ω, the overlap ratio and the total number of processor cores are the same as the noise level test, i.e., $nd = 8$, $\gamma = 0.5$ and $np = 216$. The mesh for computation is $37 \times 37 \times 37$ in space and $M = 37$ in time. The measurement data ratio is set as $r = 1/2$, $1/3$ and $1/4$ respectively, which means the measurements on each subdomain are $18 \times 18 \times 18$, $12 \times 12 \times 12$ and $9 \times 9 \times 9$. The noise level is fixed at 1% for all examples. The regularization parameters are also chosen heuristically. Homogeneous boundary condition in (21) or the estimation-correction boundary condition in Fig. 2 are applied on the artificial boundaries. The numerical results with the homogeneous and the estimation-correction boundary conditions are denoted by “S1” and “S2” respectively. For the estimation-correction process, the maximum GMRES iteration number for estimation and correction stage are 30 and 200 respectively.

The average number of GMRES iterations, the average computing time, and the reconstruction errors are summarized in Table 3. It is noted that for the estimation-correction process, the GMRES iteration number is the sum of both the estimation stage and the correction stage, the computing time consists of three parts: the solving time for the above two stages, as well as the communication and interpolation time for correcting the artificial boundary conditions. It is shown that the number of GMRES iterations, the computing time and the error increase accordingly with the decreases of measurement data. The iteration number and the computing time of “S2” are slightly larger than those of “S1” for measurement data ratio $r = 1/2$, but significantly outperforms the results of “S1” for $r = 1/3$ and $1/4$. The summation error of “S2” keeps at almost the same level as that of “S1” for $r = 1/2$; nevertheless for $r = 1/3$ and $1/4$ the error reduces as much as about 0.02 comparing to that of “S1”. In our numerical experiments the communication and interpolation for correcting the artificial boundary conditions cost about 0.3 sec for Example 1-3, which is quite a small proportion less than 1% to the total computing time of “S2”. Therefore one can conclude that when the measurement data is sufficient, the estimation-correction process has little advantage than the homogeneous boundary condition; however when the measurement data is insufficient, the estimation-correction process can significantly improve the reconstruction accuracy and achieve faster convergence with a little extra computation cost.

For the second part, we intend to figure out how the reconstruction error changes with the overlapping size between neighboring subdomains when applying the above two boundary conditions. Ω is still partitioned into $nd = 8$ subdomains, the mesh on each subdomain is refined to $41 \times 41 \times 41$ and the time step $M = 41$. The measurement data ratio is $r = 1/4$. The total number of processor cores is $np = 512$, which implies each subproblem is solved by 64 processor cores. We set the overlap ratio as $\gamma = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8$ and 0.9, and plot the corresponding summation error E in Fig. 8. As observed from Fig. 8, when the overlap ratio increases from 0.2 to 0.5, the errors decrease significantly for Example 1-3; then the errors reduce slightly for γ enlarging from 0.5 to 0.7 and keep steady for $\gamma \geq 0.7$. These results show that the appropriate overlap ratio is around 0.5. Furthermore we notice that the errors for the estimation-correction process (denoted by blue star line)

Table 3
Measurement data test with homogeneous boundary condition (denoted by "S1") and the estimation-correction boundary condition (denoted by "S2").

r	Ex1-S1				Ex1-S2			
	β	N_{its}	T_{GMRES}	E	β	N_{its}	T_{GMRES}	E
$\frac{1}{2}$	10^{-4}	81	30.64	0.040	10^{-4}	83	31.19	0.038
$\frac{1}{3}$	10^{-4}	111	39.73	0.057	10^{-4}	89	33.15	0.044
$\frac{1}{4}$	10^{-4}	142	48.92	0.069	10^{-4}	106	38.10	0.049
r	Ex2-S1				Ex2-S2			
	β	N_{its}	T_{GMRES}	E	β	N_{its}	T_{GMRES}	E
$\frac{1}{2}$	10^{-4}	82	30.85	0.035	10^{-4}	83	31.33	0.035
$\frac{1}{3}$	10^{-4}	111	39.82	0.048	10^{-4}	89	33.24	0.035
$\frac{1}{4}$	10^{-4}	142	48.97	0.058	10^{-4}	106	38.20	0.040
r	Ex3-S1				Ex3-S2			
	β	N_{its}	T_{GMRES}	E	β	N_{its}	T_{GMRES}	E
$\frac{1}{2}$	10^{-5}	51	21.58	0.106	10^{-5}	54	22.45	0.103
$\frac{1}{3}$	10^{-4}	111	39.59	0.123	10^{-5}	88	32.91	0.116
$\frac{1}{4}$	10^{-4}	142	48.83	0.136	10^{-4}	106	37.89	0.126

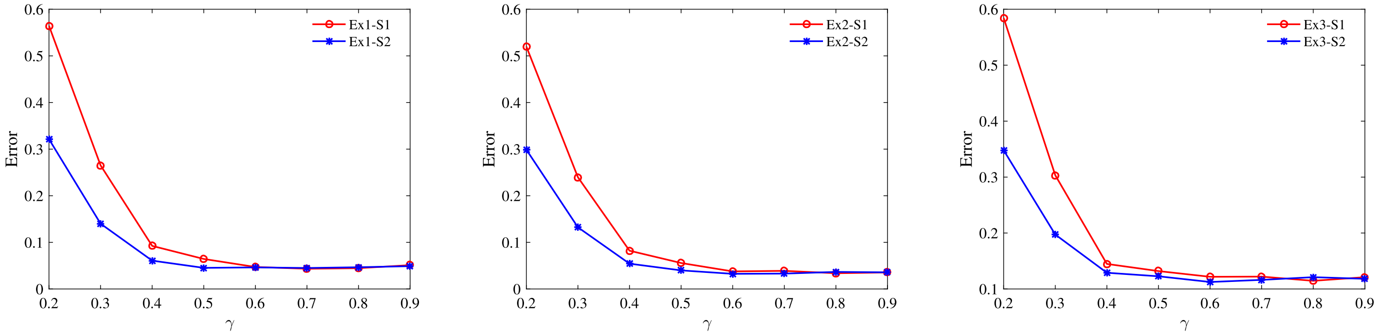

Fig. 8. Overlapping size test with two different artificial boundary conditions. The red circle line stands for the results with the homogeneous boundary condition, and the blue star line stands for the results with the estimation-correction boundary condition. (Left: Example 1; middle: Example 2; right: Example 3.)

Table 4
The parallel performance with fixed total mesh and different number of subdomains nd. The number of processor cores used in the computation is $np = 2048$. "Sp." stands for the speedup.

nd	Ex1				Ex2				Ex3			
	N_{its}	T_{GMRES}	E	Sp.	N_{its}	T_{GMRES}	E	Sp.	N_{its}	T_{GMRES}	E	Sp.
1	140	21.85	0.062	1	139	19.18	0.069	1	141	20.70	0.110	1
8	106	14.50	0.070	1.51	107	14.47	0.055	1.33	105	14.20	0.119	1.46
64	70	9.62	0.064	2.27	71	9.60	0.062	2.00	53	7.84	0.113	2.64

are always smaller than those of the homogeneous boundary conditions (denoted by red circle line) for $\gamma \leq 0.6$ and have no evident difference when $\gamma > 0.6$. It indicates that when the overlap is insufficiently small, the estimation-correction process can substantially reduce the reconstruction error.

4.4. Parallel efficiency test

In this section we show the computing efficiency with different number of subdomains nd in the first-level domain decomposition. Ω is divided into $nd = 1$, 8 or 64 subdomains of equal volume. The mesh on each subdomain is $97 \times 97 \times 97$ for $nd = 1$, $49 \times 49 \times 49$ for $nd = 8$ and $25 \times 25 \times 25$ for $nd = 64$ respectively, the time step $M = 41$. As observed the total problem size for the three cases is fixed to a total space mesh $97 \times 97 \times 97$ and $np = 2048$ processor cores are used for computing all examples. The noise level is $\epsilon = 1\%$ and the measurement ratio is $r = 1/2$. The summation error, the average GMRES iteration number, the average computing time and the speedup (denoted by Sp.) of $nd = 8$ and 64 are compared with the case of $nd = 1$, i.e., the inverse source problem is solved traditionally without domain decomposition at the first level. As observed from Table 4, the iteration number and the computing time reduce significantly as the number of subdomains increases, and the reconstruction errors are still satisfactory. For the same problem size and the same number of processor cores, the computing time for the cases of $nd = 8$ and $nd = 64$ are 1.43 and 2.30 times respectively faster than that of $nd = 1$ in average. As all the KKT subsystems are solved in parallel and independently, faster convergence is achieved than solving the all-at-once large system.

Furthermore we compare the weak scalability of the proposed two-level domain decomposition method with the traditional method. Firstly we divide Ω into $nd = 1$, 8, 27 or 64 subdomains, on each subdomain the mesh is set as $41 \times 41 \times 41$ and the time step $M = 41$. Here the problem size of each subdomain is the same and 64 processor cores are used for the computation of each subproblem, thus the total num-

Table 5
Weak scalability test with $nd \geq 1$ and $nd = 1$, "Eff." stands for the parallel efficiency.

np	Ex1 ($nd \geq 1$)					Ex1 ($nd = 1$)			
	nd	N_{its}	T_{GMRES}	E	Eff.	N_{its}	T_{GMRES}	E	Eff.
64	1	71	19.14	0.040	100%	71	19.14	0.040	100%
512	8	84	22.49	0.042	85.11%	123	32.33	0.040	59.21%
1728	27	100	26.06	0.047	73.46%	218	59.52	0.046	32.17%
4096	64	155	37.66	0.047	50.83%	-	-	-	-
np	Ex2 ($nd \geq 1$)					Ex2 ($nd = 1$)			
	nd	N_{its}	T_{GMRES}	E	Eff.	N_{its}	T_{GMRES}	E	Eff.
64	1	72	19.38	0.032	100%	72	19.38	0.032	100%
512	8	84	22.46	0.036	86.31%	124	33.38	0.038	58.07%
1728	27	102	26.37	0.043	73.51%	220	60.01	0.039	32.31%
4096	64	130	32.36	0.047	59.91%	-	-	-	-
np	Ex3 ($nd \geq 1$)					Ex3 ($nd = 1$)			
	nd	N_{its}	T_{GMRES}	E	Eff.	N_{its}	T_{GMRES}	E	Eff.
64	1	46	14.02	0.110	100%	46	14.02	0.110	100%
512	8	52	15.84	0.103	88.47%	124	32.77	0.103	42.78%
1728	27	62	17.79	0.102	78.81%	222	59.48	0.104	23.57%
4096	64	80	21.47	0.093	65.28%	-	-	-	-

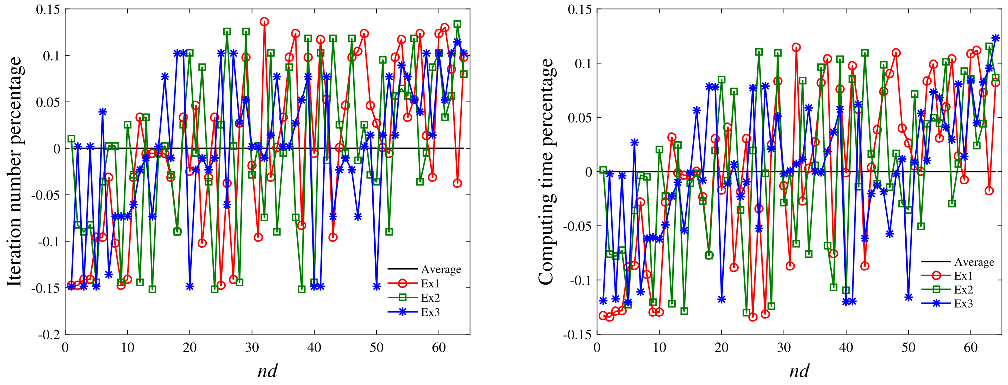

Fig. 9. Performance comparison of each subproblem versus the mean values ($nd = 64$). Left: Iteration number, right: Computing time.

ber of processor cores used for $nd = 1$, 8, 27 and 64 are $np = 64$, 512, 1728 and 4096 respectively. Then for demonstrating the weak scalability of the traditional method, we solve the problem with refined meshes $81 \times 81 \times 81$, $121 \times 121 \times 121$ and $161 \times 161 \times 161$ without the first-level domain decomposition, and $np = 512$, 1728 and 4096 processor cores are used respectively. The problem scale and the number of processor cores are correspondingly the same with the cases for $nd = 8$, 27 and 64. It is noted that we tune the GMRES stopping criterion $rtol$ and the regularization parameters so as to make sure the reconstruction errors do not raise significantly for increasing problem size and maintain at almost the same level for all experiments. The number of GMRES iterations, the computing time, the summation error and the parallel efficiency (denoted by Eff.) are shown in Table 5.

It is observed from Table 5 that for the proposed two-level domain decomposition method ($nd \geq 1$), the average number of GMRES iterations grows with nd, meanwhile the computing time increases mildly, the parallel efficiency is about 58.67% in average comparing $nd = 64$ to $nd = 1$. For the classical method without domain decomposition at the first level ($nd = 1$), when the mesh is refined from $41 \times 41 \times 41$ to $121 \times 121 \times 121$, the parallel efficiency is reduced rapidly to 32.17%, 32.31% and 23.57% respectively for Example 1, 2 and 3. In comparison the corresponding efficiency for the two-level domain decomposition method is 73.46%, 73.51% and 78.81% respectively (see Table 5 for the rows with $np = 1728$). When the problem size is enlarged to the mesh $161 \times 161 \times 161$ with $nd = 1$, system errors prompt and the cluster fails to solve the system of about 5.13×10^8 unknowns, while the proposed two-level algorithm scales weakly to the same mesh with reasonable results obtained (see Table 5 for the rows with $np = 4096$). In this situation dividing the problem into smaller subproblems is necessary. In summary, we see that better weak scalability is achieved with the proposed method.

Finally, to investigate the performance balance between subdomains, for the case of $nd = 64$, we compare the iteration number of each subproblem versus the average iteration number and the individual computing time versus the average computing time in Fig. 9. Taking the average GMRES iteration number and the average computing time as benchmarks, the results show that the iteration number variates in the ranges $[-14.75\%, 13.66\%]$, $[-15.17\%, 13.36\%]$ and $[-14.83\%, 11.47\%]$, and the computing time variates in the ranges $[-13.42\%, 11.44\%]$, $[-13.03\%, 11.56\%]$ and $[-12.05\%, 12.32\%]$ for Example 1, 2 and 3 respectively. The variance of the computing time is slightly smaller than that of the iteration number, and the computing performance differences among subproblems are acceptable.

5. Some final remarks

We develop a two-level domain decomposition method for the 3D unsteady source identification problem. When the original inverse problem is transformed into smaller independent subproblems defined on overlapping subdomains, forward model errors arise from unknown inner boundary conditions. To overcome this difficulty, appropriate overlap is necessary and two types of Dirichlet conditions are imposed on the artificial boundaries for the forward parabolic equation. Instead of solving the space-time fully coupled KKT system, the resulting KKT subsystems are efficiently solved in parallel by a preconditioned GMRES method with faster convergence achieved. Numerical tests show that the proposed algorithm is robust with different noise level, large terminal time, the amount of measurement data, the overlapping size and the number of subdomains. Moreover the reconstruction accuracy is comparable with the classical approach without decomposition at the optimization level, especially when the artificial boundary conditions are decided by an estimation-correction process. Computing efficiency tests show that the first-level domain decomposition can speedup the computation for fixed problem size, and lead to a much better weak scalability with up to 4096 processor cores. The two-level domain decomposition method is quite adaptable to large-scale problems and it has the potential to do fast and reliable simulations for space-time dependent inverse problems.

Acknowledgements

The authors would like to thank the referees for the valuable comments that helped improve this paper. This work was supported by the Guangdong Provincial Natural Science Foundation (2020A1515010704, 2021A1515012366).

References

[1] H.W. Engl, M. Hanke, A. Neubauer, Regularization of Inverse Problems, Vol. 375, Springer Science & Business Media, 1996.
[2] A.A. Samarskii, P.N. Vabishchevich, Numerical Methods for Solving Inverse Problems of Mathematical Physics, Vol. 52, Walter de Gruyter, 2008.
[3] L. Yang, Z.-C. Deng, J.-N. Yu, G.-W. Luo, Optimization method for the inverse problem of reconstructing the source term in a parabolic equation, Math. Comput. Simul. 80 (2) (2009) 314–326.
[4] A. Ashyralyev, A.U. Sazaklioglu, Investigation of a time-dependent source identification inverse problem with integral overdetermination, Numer. Funct. Anal. Optim. 38 (10) (2017) 1276–1294.
[5] D.N. Hào, B.V. Huong, N.T.N. Oanh, P.X. Thanh, Determination of a term in the right-hand side of parabolic equations, J. Comput. Appl. Math. 309 (2017) 28–43.
[6] D.N. Hào, T.N.T. Quyen, N.T. Son, Convergence analysis of a Crank–Nicolson Galerkin method for an inverse source problem for parabolic equations with boundary observations, Appl. Math. Optim. 84 (2) (2021) 2289–2325.
[7] T. Cheng, J. Hu, D. Jiang, Simultaneous identification of convection velocity and source strength in a convection–diffusion equation, Appl. Anal. 99 (12) (2020) 2170–2189.
[8] O.Y. Imanuvilov, M. Yamamoto, Lipschitz stability in inverse parabolic problems by the Carleman estimate, Inverse Probl. 14 (5) (1998) 1229.
[9] A. Prilepko, D. Tkachenko, Inverse problem for a parabolic equation with integral overdetermination, J. Inverse Ill-Posed Probl. 11 (2) (2003).
[10] A.I. Prilepko, V.L. Kamynin, A.B. Kostin, Inverse source problem for parabolic equation with the condition of integral observation in time, J. Inverse Ill-Posed Probl. 26 (4) (2018) 523–539.
[11] X. Liu, Z. Zhai, Inverse modeling methods for indoor airborne pollutant tracking: literature review and fundamentals, Indoor Air 17 (6) (2007) 419–438.
[12] E. Lushi, J.M. Stockie, An inverse Gaussian plume approach for estimating atmospheric pollutant emissions from multiple point sources, Atmos. Environ. 44 (8) (2010) 1097–1107.
[13] S. Alapati, Z. Kabala, Recovering the release history of a groundwater contaminant using a non-linear least-squares method, Hydrol. Process. 14 (6) (2000) 1003–1016.
[14] A. Hamdi, I. Mahfoudhi, Inverse source problem in a one-dimensional evolution linear transport equation with spatially varying coefficients: application to surface water pollution, Inverse Probl. Sci. Eng. 21 (6) (2013) 1007–1031.
[15] V. Akçelik, G. Biros, O. Ghattas, K.R. Long, B. van Bloemen Waanders, A variational finite element method for source inversion for convective–diffusive transport, Finite Elem. Anal. Des. 39 (8) (2003) 683–705.
[16] H.P. Flath, L.C. Wilcox, V. Akçelik, J. Hill, B. van Bloemen Waanders, O. Ghattas, Fast algorithms for Bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial Hessian approximations, SIAM J. Sci. Comput. 33 (1) (2011) 407–432.
[17] J.-F. Wong, P. Yuan, A FE-based algorithm for the inverse natural convection problem, Int. J. Numer. Methods Fluids 68 (1) (2012) 48–82.
[18] M. Hinze, T.N.T. Quyen, Finite element approximation of source term identification with TV-regularization, Inverse Probl. 35 (12) (2019) 124004.
[19] S. Mo, N. Zabaras, X. Shi, J. Wu, Deep autoregressive neural networks for high-dimensional inverse problems in groundwater contaminant source identification, Water Resour. Res. 55 (5) (2019) 3856–3881.
[20] A. Hasanov, B. Mukanova, Fourier collocation algorithm for identifying the spacewise-dependent source in the advection–diffusion equation from boundary data measurements, Appl. Numer. Math. 97 (2015) 1–14.
[21] S.M. Gorelick, B. Evans, I. Remson, Identifying sources of groundwater pollution: an optimization approach, Water Resour. Res. 19 (3) (1983) 779–790.
[22] V. Akçelik, G. Biros, A. Draganescu, J. Hill, O. Ghattas, B.V.B. Waanders, Dynamic data-driven inversion for terascale simulations: real-time identification of airborne contaminants, in: SC'05: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing, IEEE, 2005.
[23] K. Cao, D. Lesnic, Simultaneous reconstruction of the spatially-distributed reaction coefficient, initial temperature and heat source from temperature measurements at different times, Comput. Math. Appl. 78 (10) (2019) 3237–3249.
[24] G. Li, Y. Wang, A regularizing trust region algorithm for nonlinear ill-posed problems, Inverse Probl. Sci. Eng. 14 (8) (2006) 859–872.
[25] D. Jiang, Z. Li, Y. Liu, M. Yamamoto, Weak unique continuation property and a related inverse source problem for time-fractional diffusion-advection equations, Inverse Probl. 33 (5) (2017) 055013.
[26] Z. Ruan, S. Zhang, W. Zhang, Numerical solution of time-dependent component with sparse structure of source term for a time fractional diffusion equation, Comput. Math. Appl. 77 (5) (2019) 1408–1422.
[27] N.M. Dien, D.N.D. Hai, T.Q. Viet, D.D. Trong, On Tikhonov's method and optimal error bound for inverse source problem for a time-fractional diffusion equation, Comput. Math. Appl. 80 (1) (2020) 61–81.
[28] J. Yu, F. Li, S. Yu, Z. Wu, The approximate solutions to source inverse problem of 1-D convection–diffusion equation by LS-SVM, Inverse Probl. Sci. Eng. 26 (5) (2018) 677–690.
[29] H. Li, J. Zhang, Fast source term estimation using the PGA-NM hybrid method, Eng. Appl. Artif. Intell. 62 (2017) 68–79.
[30] X. Deng, X.-C. Cai, J. Zou, Two-level space–time domain decomposition methods for three-dimensional unsteady inverse source problems, J. Sci. Comput. 67 (3) (2016) 860–882.
[31] X. Deng, X.-C. Cai, J. Zou, A parallel space-time domain decomposition method for unsteady source inversion problems, Inverse Probl. Imaging 9 (4) (2015) 1069.
[32] L.C. Evans, Partial Differential Equations, second edition, American Mathematical Society, 2010.
[33] K. Ito, B. Jin, Inverse problems: Tikhonov theory and algorithms, World Scientific, 2014.
[34] X. Deng, Z.-J. Liao, X.-C. Cai, A parallel multilevel domain decomposition method for source identification problems governed by elliptic equations, J. Comput. Appl. Math. 392 (2021) 113441.
[35] D. Jiang, H. Feng, J. Zou, Overlapping domain decomposition methods for linear inverse problems, Inverse Probl. Imaging 9 (1) (2015) 163.
[36] S. Balay, S. Abhyankar, M.F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, W.D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M.G. Knepley, F. Kong, S. Kruger, D.A. May, L.C. McInnes, R.T. Mills, L. Mitchell, T. Munson, J.E. Roman, K. Rupp, P. Sanan, J. Sarich, B.F. Smith, S. Zampini, H. Zhang, J. Zhang, PETSc Users Manual, Tech. Rep. ANL-21/39 - Revision 3.16, Argonne National Laboratory, 2021.