A PARALLEL SPACE-TIME DOMAIN DECOMPOSITION METHOD FOR UNSTEADY SOURCE INVERSION PROBLEMS

Xiaomao Deng

Laboratory for Engineering and Scientific Computing
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
Shenzhen, Guangdong 518055, China

Xiao-Chuan Cai

Department of Computer Science
University of Colorado
Boulder, CO 80309, USA

Jun Zou*

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

(Communicated by Samuli Siltanen)

Abstract. In this paper, we propose a parallel space-time domain decomposition method for solving an unsteady source identification problem governed by the linear convection-diffusion equation. Traditional approaches require to solve repeatedly a forward parabolic system, an adjoint system and a system with respect to the unknown sources. The three systems have to be solved one after another. These sequential steps are not desirable for large scale parallel computing. A space-time restrictive additive Schwarz method is proposed for a fully implicit space-time coupled discretization scheme to recover the time-dependent pollutant source intensity functions. We show with numerical experiments that the scheme works well with noise in the observation data. More importantly it is demonstrated that the parallel space-time Schwarz preconditioner is scalable on a supercomputer with over 10^3 processors, thus promising for large scale applications.

1. **Introduction.** Pollutant source inversion problems have wide applications in, for example, the detection and monitoring of indoor and outdoor air pollution, underground water pollution, etc. In the last several decades, physical, chemical and biological technologies have been developed to identify different types of sources [3, 45, 46]. In this paper, assuming the pollutant concentration data is measured by distributed sensors, we reconstruct the source intensities numerically using noise-contaminated data. Like all inverse problems, such a reconstruction problem is ill-posed in the sense of Hadamard [14, 34, 41]. The lack of stability with respect to

the measurement data is a major issue, which means that small noise in the data may lead to significant changes in the reconstructed source strength. This problem has attracted much attention, and various methods have been developed, including both deterministic and statistical methods [27, 37]. Among the deterministic methods, quasi-explicit reconstruction formulas are available for one-dimensional source location recovery problems [19, 20]; and quasi-reversibility methods can be used to retrace the pollutant history as in [36]; optimization based methods are also widely used [2, 22, 23, 35, 40, 42]. By reformulating an inverse problem into an output least-squares PDE-constrained optimization problem complemented with Tikhonov regularization, classical optimization methods such as regression methods [17], linear and nonlinear programming methods [17], linear and nonlinear conjugate gradient methods [1, 40], Newton type methods, etc. can be used to obtain the approximate solutions. These methods can be categorized as sequential quadratic programming (SQP) methods. Reduced space SQP methods decouple the system and iteratively update the state variable, the adjoint variable and the optimization variables by solving each subsystem in a sequential order. In some sense this is a block Gauss-Seidel iteration with three large blocks. Such methods require less memory due to the reduced subproblem size but the number of outer iterations for a specified accuracy grows quickly with the increase of the optimization variables, thus they are not ideal for supercomputers with a large number of processors.

We introduce in this paper a full space approach that does not have the three large sequential steps as in the reduced space approaches. Similar approaches have been applied to flow control problems in [31]. The full space method solves the state variable, adjoint variable and the optimization variables simultaneously, thus avoids repeatedly solving the subsystems. However the fully coupled system is several times larger in size and more ill-conditioned, direct methods such as Gaussian elimination or LU factorization as well as the classical iterative methods such as the Jacobi method, the Gauss-Seidel method are not suitable. To ease the difficulty of solving the large system, a preconditioned Krylov subspace technique is considered to reduce the condition number and the computing time significantly [9, 44].

The inverse problem of recovering the pollutant source intensity functions can be reformulated into a PDE-constrained optimization problem. In this paper, we derive its continuous Karush-Kuhn-Tucker (KKT) system [24], including the state equation, the adjoint equation and other derivative equations with respect to each unknown source intensity. Two main challenges of the problem lie in that firstly the adjoint equation needs the final state of the pollutant source distribution, which implies that the state equation and the adjoint equation should be solved in a sequential order; secondly the time marching of the unsteady problem is directional and sequential, thus difficult to break down into parallel steps. For unsteady PDE-constrained optimization problems, a steady state optimization subproblem is solved at each time step [44]. And in [18], a block time-marching method is used to reduce the number of sequential steps and increase the degree of parallelism. In this paper, we propose a fully coupled space-time domain decomposition method that couples the time with the space domain and decomposes the "space-time" domain into subdomains, then apply an additive Schwarz preconditioned Krylov subspace technique to solve the "space-time" problem. Our algorithm is fully parallel in space and time, avoids the sequential time marching steps, and does not need to solve optimization subproblems. As far as we know, no published work has achieved such a degree of parallelism for time-dependent inverse problems.

The rest of this paper is arranged as follows. The mathematical model and its corresponding optimization functional, and the derivation of the KKT system are formulated in Section 2. The discretization of the KKT system is given in Section 3. The parallel algorithm for solving the KKT system is proposed in Section 4. Some numerical experiments are shown in Section 5 and concluding remarks are given in Section 6.

2. **Model formulation.** We consider a flow domain $\Omega \in \mathbf{R}^2$ in which several point pollutant sources are present. The distribution of the pollutant concentration is denoted by $C(\mathbf{x}, t)$ at location $\mathbf{x}$ and time t. The transport process is modeled by the following convection-diffusion equation [3, 32]:

$$(1) \quad \frac{\partial C}{\partial t} = \nabla \cdot (a(\mathbf{x})\nabla C) - \nabla \cdot (\mathbf{v}(\mathbf{x})C) + \sum_{i=1}^{s} \delta(\mathbf{x} - \mathbf{x}_i^*) f_i(t), \ 0 < t < T, \ \mathbf{x} \in \Omega,$$

where $f_i(t)$ is the temporal intensity of the ith source at location $\mathbf{x}_i^*$, $i = 1, \cdots, s$, s is the number of sources, $a(\mathbf{x})$ and $\mathbf{v}(\mathbf{x})$ are the diffusive and convective coefficient. $\delta(\cdot)$ is the Dirac delta distribution [5]. The model is complemented by the following boundary conditions

$$(2) \quad C(\mathbf{x}, t) = p(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma_1; \quad a(\mathbf{x})\frac{\partial C}{\partial \mathbf{n}} = q(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma_2$$

and the initial condition

$$(3) \quad C(\mathbf{x}, 0) = C_0(\mathbf{x}), \quad \mathbf{x} \in \Omega,$$

where Γ_1 and Γ_2 cover the physical boundary $\overline{\partial \Omega} = \overline{\Gamma}_1 \bigcup \overline{\Gamma}_2$, $p(\mathbf{x}, t)$ and $q(\mathbf{x}, t)$ are given functions for Dirichlet and Neumann boundary condition respectively. If the source locations $\mathbf{x}_i^*$ $(i = 1, \cdots, s)$ and the corresponding time-dependent intensities $f_i(t)$ $(i = 1, \cdots, s)$ in (1) are all known, then the distribution of the pollutant concentration $C(\mathbf{x}, t)$ can be obtained by solving the convection-diffusion equation (1)-(3). This is usually called a forward or direct problem. In this paper, we are concerned about the inverse problem, that is, using the noise-contaminated data $C^\varepsilon(\mathbf{x}, t)$ (ε is the noise level) of the concentration $C(\mathbf{x}, t)$ in Ω at terminal time T to recover the source intensity functions $f_i(t)$ $(i = 1, \cdots, s)$. In practice, the data $C^\varepsilon(\mathbf{x}, t)$ is measured by a sensor network placed at some discrete points inside the domain Ω [26, 30]. A discussion of the sensor network can be found in [26], but we shall assume that the measurement data is available here at a set of uniformly distributed sensors inside Ω.

The Tikhonov optimization algorithm is popular for time-dependent parameter identification problems [22, 23]. The main ingredient of the algorithm includes reformulating the reconstruction process as the minimization of the following functional:

$$(4) \quad J(\mathbf{f}) = \frac{1}{2} \int_\Omega (C(\mathbf{x}, T) - C^\varepsilon(\mathbf{x}, T))^2 \, d\mathbf{x} + N_\beta(\mathbf{f}),$$

where $\mathbf{f} = (f_1, f_2, \cdots, f_s)^T$, and $N_\beta(\mathbf{f})$ denotes some Tikhonov regularization. Possible choices for the regularizations include L^2, H^1 and BV regularizations. Here we consider a combination of the L^2 and H^1 regularizations in the following form

$$(5) \quad N_\beta(\mathbf{f}) = \sum_{i=1}^{s} \frac{\beta_1^i}{2} \int_0^T (f_i(t))^2 dt + \sum_{i=1}^{s} \frac{\beta_2^i}{2} \int_0^T |f_i'(t)|^2 dt,$$

where β_1^i, β_2^i, $i = 1, \cdots, s$, are the L^2 or H^1 regularization parameters for the source intensity $f_1(t), \cdots, f_s(t)$ respectively. The minimization of the functional (4) is subject to the constraints that $C(\mathbf{x}, t)$ satisfies the state equation (1) with the boundary conditions (2) and the initial condition (3). This has transformed the original inverse source problem into a PDE-constrained optimization problem.

Two kinds of approaches for the optimization problem (4) are available, the discretize-then-optimize approach and the optimize-then-discretize approach. The solutions from both approaches are credible, although they are not necessarily the same [31]. We shall use the optimize-then-discretize approach in this work.

Let $W^{1,p}(\Omega)$ and $W^{1,q}(\Omega)$ be standard Sobolev spaces with $p, q > 0$ such that $1/p + 1/q = 1$ and $p > 2$, $q < 2$. We formally write (1) as an operator equation $L(C, \mathbf{f}) = 0$, then introduce a corresponding Lagrange multiplier $G \in W^{1,p}(\Omega)$ and the following Lagrange functional [2, 22, 23]:

$$(6) \qquad \mathcal{J}(C, \mathbf{f}, G) = \frac{1}{2} \int_\Omega (C(\mathbf{x}, T) - C^\varepsilon(\mathbf{x}, T))^2 d\mathbf{x} + N_\beta(\mathbf{f}) + (G, L(C, \mathbf{f})),$$

where G is the Lagrange multiplier or the adjoint variable, and $(G, L(C, \mathbf{f}))$ stands for the dual product.

Taking the variations of (6) with respect to G, C and f_i, $i = 1, \cdots, s$, a system of partial differential equations is derived to characterize the first-order optimality conditions for this optimization problem (6). They are the so-called Karush-Kuhn-Tucker (KKT) optimality conditions [24]. It has been verified that the minimization problem (4) is equivalent to solving the KKT system of the Lagrangian functional $\mathcal{J}(C, \mathbf{f}, G)$ in [11]. The three sets of equations in the KKT system are obtained as follows:

(a) The Gâteaux derivative of $\mathcal{J}$ with respect to G at direction v is given by

$$\mathcal{J}_G(C, \mathbf{f}, G)v = (v, L(C, \mathbf{f}))$$
$$= \left(\frac{\partial C}{\partial t}, v \right) + (a\nabla C, \nabla v) + (\nabla \cdot (\mathbf{v}C), v)$$
$$- \sum_{i=1}^{s} v(\mathbf{x}_i^*) f_i(t) - \langle q, v \rangle_{\Gamma_2}$$

for all $v \in W^{1,p}(\Omega)$.

(b) The Gâteaux derivative of $\mathcal{J}$ in (6) with respect to C at direction $w \in W^{1,q}(\Omega)$ is given by

$$(7) \qquad \begin{aligned} \mathcal{J}_C(C, \mathbf{f}, G)w &= \int_\Omega (C(\mathbf{x}, T) - C^\varepsilon(\mathbf{x}, T))w d\mathbf{x} \\ &+ \int_0^T \int_\Omega G \left(\frac{\partial w}{\partial t} - \nabla \cdot (a(\mathbf{x})\nabla w) + \nabla \cdot (\mathbf{v}(\mathbf{x})w) \right) d\mathbf{x}dt. \end{aligned}$$

For convenience, we write

$$\tilde{L}w := \frac{\partial w}{\partial t} - \nabla \cdot (a(\mathbf{x})\nabla w) + \nabla \cdot (\mathbf{v}(\mathbf{x})w),$$

and obtain by integrating by part for the second term of (7) that

$$(G, \tilde{L}w) = \int_0^T \int_\Omega G \left(\frac{\partial w}{\partial t} - \nabla \cdot (a(\mathbf{x})\nabla w) + \nabla \cdot (\mathbf{v}(\mathbf{x})w) \right) d\mathbf{x}dt$$

$$= \int_\Omega Gw|_0^T d\mathbf{x} - \int_0^T \int_\Omega w\frac{\partial G}{\partial t}d\mathbf{x}dt - \int_0^T \int_{\partial\Omega} a(\mathbf{x})\frac{\partial w}{\partial \mathbf{n}}Gd\Gamma dt$$

$$+ \int_0^T \int_\Omega a(\mathbf{x})\nabla w \cdot \nabla G d\mathbf{x}dt + \int_0^T \int_\Omega \nabla \cdot (\mathbf{v}(\mathbf{x})w)G d\mathbf{x}dt$$

$$= \int_\Omega Gw|_0^T d\mathbf{x} + \int_0^T \int_{\partial\Omega} \left(-a(\mathbf{x})\frac{\partial w}{\partial \mathbf{n}}G\right) d\Gamma dt$$

$$+ \int_0^T \int_\Omega \left(-w\frac{\partial G}{\partial t} + a(\mathbf{x})\nabla w \cdot \nabla G + \nabla \cdot (\mathbf{v}(\mathbf{x})w)G\right) d\mathbf{x}dt .$$

Then applying the boundary and initial conditions of w, i.e. $w = 0$ on Γ_1 and $a(\mathbf{x})\frac{\partial w}{\partial \mathbf{n}} = 0$ on Γ_2, $w(\mathbf{x}, 0) = 0$, we derive

$$(G, \tilde{L}w) = \int_\Omega G(\mathbf{x}, T)w(\mathbf{x}, T)d\mathbf{x} + \int_0^T \int_{\Gamma_1} \left(-a(\mathbf{x})\frac{\partial w}{\partial \mathbf{n}}G\right) d\Gamma dt$$

$$+ \int_0^T \int_\Omega \left(-w\frac{\partial G}{\partial t} + a(\mathbf{x})\nabla w \cdot \nabla G + \nabla \cdot (\mathbf{v}(\mathbf{x})w)G\right) d\mathbf{x}dt.$$

Now noting the arbitrariness of w, we can deduce the adjoint system for the Lagrange multiplier G, namely $G(\mathbf{x}, T) = 0$ for $\mathbf{x} \in \Omega$, $G(\mathbf{x}, t) = 0$ on Γ_1 and $G(\mathbf{x}, t)$ satisfies

$$(8) \quad -(G_t, w) + (a\nabla G, \nabla w) + (\nabla \cdot (\mathbf{v}w), G) = -(\delta(t - T)(C(\cdot, t) - C^\varepsilon(\cdot, t)), w)$$

for all $w \in W^{1,q}(\Omega)$ such that $w = 0$ on Γ_1, where $\delta(t - T)$ is the Dirac delta distribution at $t = T$.

(c) The Gâteaux derivative of $\mathcal{J}$ in (6) with respect to f_i at direction $g \in H^1(0, T)$ is given by

$$\mathcal{J}_{f_i}(C, \mathbf{f}, G)g = \beta_1^i \int_0^T f_i(t)g(t)dt + \beta_2^i \int_0^T f_i'(t)g'(t)dt$$

$$(9) \qquad\qquad - \int_0^T \int_\Omega (G(\mathbf{x}, t), \delta(\mathbf{x} - \mathbf{x}_i^*)g(t))d\mathbf{x}dt$$

$$= \int_0^T (\beta_1^i f_i(t) - G(\mathbf{x}_i^*, t))g(t)dt + \beta_2^i \int_0^T f_i'(t)g'(t)dt.$$

Putting (a)-(c) together, the KKT system is formulated as follows:

$$(10) \qquad \begin{cases} \mathcal{J}_G(C, \mathbf{f}, G)v = 0 \\ \mathcal{J}_C(C, \mathbf{f}, G)w = 0 \\ \mathcal{J}_{f_i}(C, \mathbf{f}, G)g = 0, \quad i = 1, \cdots, s, \end{cases}$$

that is, for any $v \in W^{1,p}(\Omega)$ and $w \in W^{1,q}(\Omega)$, we have the following coupled system:

$$(11) \quad \begin{cases} \left(\dfrac{\partial C}{\partial t}, v\right) + (a\nabla C, \nabla v) + (\nabla \cdot (\mathbf{v}C), v) - \displaystyle\sum_{i=1}^s v(\mathbf{x}_i^*)f_i(t) - \langle q, v\rangle_{\Gamma_2} = 0 \\ -\left(\dfrac{\partial G}{\partial t}, w\right) + (a(\mathbf{x})\nabla G, \nabla w) + (\nabla \cdot (\mathbf{v}(\mathbf{x})w), G) \\ \quad + (\delta(t - T)(C(\cdot, t) - C^\varepsilon(\cdot, t)), w) = 0 \\ -(G(\mathbf{x}_i^*, \cdot), g) + \beta_1^i(f_i, g) + \beta_2^i(f_i', g') = 0, \quad i = 1, \cdots, s \end{cases}$$

with $C(\mathbf{x}, 0) = C_0(\mathbf{x})$, $G(\mathbf{x}, T) = 0$. The rest of the paper is devoted to solving (11) as a coupled space-time system. It is noted that the first equation in (11) is the state equation, and the second equation is the adjoint equation, and the last set of equations are elliptic equations with respect to each unknown source intensity.

3. **Finite element discretization.** Let $\mathcal{T}^h$ be a triangulation of Ω with triangular elements, then we define V^h as the finite element space [12] consisting of continuous piecewise linear functions on $\mathcal{T}^h$, and $\mathring{V}^h$ the subspace of V^h with functions vanishing on the Dirichlet boundary Γ_1. To fully discretize the system (11), we partition the time interval $[0, T]$ as $0 = t^0 < t^1 < \cdots < t^M = T$, with $t^n = n\tau, \tau = T/M$. Define U^τ as a piecewise linear continuous finite element space in time. For a given sequence $\{H^n(\mathbf{x}) = H(\mathbf{x}, t^n)\}$, we define the difference quotient and the averaging function respectively by

$$(12) \qquad \partial_\tau H^n(\mathbf{x}) = \frac{H^n(\mathbf{x}) - H^{n-1}(\mathbf{x})}{\tau}, \qquad \bar{H}^n = \frac{H(\mathbf{x}, t^{n-1}) + H(\mathbf{x}, t^n)}{2}.$$

Let π_h be the finite element interpolation associated with the space V^h, and $C_h^n(\mathbf{x})$ be the finite element approximation of $C(\mathbf{x}, t^n)$, then we discretize the state and adjoint equations of the system (11) by the Crank-Nicolson scheme in time and piecewise linear finite elements in space, and lastly we use piecewise linear finite element in time to discretize the equations with respect to $\mathbf{f}$. The finite element approximation of the KKT system (11) can be formulated as follows:

Find a sequence of approximations $C_h^n, G_h^n \in V^h$ ($0 \le n \le M$), $f_i^\tau \in U^\tau, i = 1, \cdots, s$, such that $C_h^0 = \pi_h C_0$, $G_h^M = \mathbf{0}$ and $C_h^n(\mathbf{x}) = \pi_h p(\mathbf{x}, t^n), G_h^n(\mathbf{x}) = \mathbf{0}$ for $\mathbf{x} \in \Gamma_1$ satisfying

(13)

$$\begin{cases} (\partial_\tau C_h^n, v_h) + (a\nabla \bar{C}_h^n, \nabla v_h) + (\nabla \cdot (\mathbf{v}\bar{C}_h^n), v_h) \\ = \sum_{i=1}^s v_h(\mathbf{x}_i^*)\bar{f}_i^n + \langle \bar{q}^n, v_h \rangle_{\Gamma_2}, \ \forall v_h \in \mathring{V}^h, \ n = 1, \cdots, M \\ -(\partial_\tau G_h^n, w_h) + (a\nabla \bar{G}_h^n, \nabla w_h) + (\nabla \cdot (\mathbf{v}w_h), \bar{G}_h^n) \\ = -\chi_n((C_h^n - C^\varepsilon), w_h), \ \forall w_h \in \mathring{V}^h, \ n = M, \cdots, 1 \\ -(G^\tau(\mathbf{x}_i^*, \cdot), g^n) + \beta_1^i(f_i^\tau, g^n) + \beta_2^i((f_i^\tau)', (g^n)') = 0, \ i = 1, \cdots, s, \ n = 0, \cdots, M, \end{cases}$$

where $\chi_n = \frac{1}{2}$ when $n = M$ and 0 when $1 \le n < M$. We denote the basis functions of finite element spaces V^h and U^τ by ϕ_i, $i = 1, \cdots, N$ and g^n, $n = 0, \cdots, M$, respectively, and introduce the following matrices:

$$\begin{aligned} A &= (a_{ij})_{i,j=1,\cdots,N}, & a_{ij} &= (a\nabla\phi_i, \nabla\phi_j) \\ B &= (b_{ij})_{i,j=1,\cdots,N}, & b_{ij} &= (\phi_i, \phi_j) \\ E &= (e_{ij})_{i,j=1,\cdots,N}, & e_{ij} &= (\nabla \cdot (\mathbf{v}\phi_i), \phi_j) \\ K &= (k_{nm})_{n,m=0,\cdots,M}, & k_{nm} &= ((g^n)', (g^m)') \\ D &= (d_{nm})_{n,m=0,\cdots,M}, & d_{nm} &= (g^n, g^m) \end{aligned}$$

and the vectors

$$\begin{aligned} C^n &= (C_1^n, C_2^n, \cdots, C_N^n)^T, & \text{for } n &= 0, \cdots, M \\ G^n &= (G_1^n, G_2^n, \cdots, G_N^n)^T, & \text{for } n &= 0, \cdots, M \\ f_k &= (f_k^0, f_k^1, \cdots, f_k^M)^T, & \text{for } k &= 1, \cdots, s \end{aligned}$$

$$r^n = (r_1^n, r_2^n, \cdots, r_N^n)^T, \quad \text{for } j = 1, \cdots, N, \ n = 1, \cdots, M \text{ with}$$

$$r_j^n = -\tau \left(\sum_{k=1}^{s} \phi_j(\mathbf{x}_k^*) \frac{(f_k^n + f_k^{n-1})}{2} + \left\langle \frac{(q^n + q^{n-1})}{2}, \phi_j \right\rangle_{\Gamma_2} \right)$$

$$g_k^* = (g_k^0, g_k^1, \cdots, g_k^M)^T, \text{ with } g_k^n = G(\mathbf{x}_k^*, t^n), \qquad \text{for } k = 1, \cdots, s, \ n = 0, \cdots, M$$

$$d = (d_1, d_2, \cdots, d_N), \text{ with } d_j = (C^\varepsilon, \phi_j) \qquad \text{for } j = 1, \cdots, N.$$

The matrix form of the KKT system is then reformulated by using the above notations as the following:

(14)
$$\begin{cases} \left(B + \dfrac{\tau}{2}(A+E)\right) C^n + \left(-B + \dfrac{\tau}{2}(A+E)\right) C^{n-1} + r^n = 0, \ n = 1, \cdots, M \\ \left(-B + \dfrac{\tau}{2}(A+E^T)\right) G^n + \left(B + \dfrac{\tau}{2}(A+E^T)\right) G^{n-1} \\ +\tau\chi_n(BC^n - d) = 0, \ n = M, \cdots, 1 \\ -Dg_k^* + (\beta_1^k D + \beta_2^k K)f_k = 0, \ k = 1, \cdots, s. \end{cases}$$

We can follow the approaches in [21, 22, 23, 42] to obtain the convergence of the discretized problem (13) to the continuous optimization problem (6).

4. A space-time domain decomposition method for the KKT system.

4.1. **Fully coupled KKT system with special ordering of unknowns.** The ordering of the unknowns for the discretized KKT system (13) has significant influence on the convergence and computing efficiency of the iterative solver. Traditional reduced space SQP methods split the system into three subsystems and solve each subsystem for C, G, and $\mathbf{f}$ one by one in sequential steps [13], in this case the unknowns are ordered physical variable by physical variable. To develop a scalable and fully coupled method for solving the KKT system, we use the so-called fully coupled ordering, the unknowns C and G are ordered mesh point by mesh point and time step by time step. At each mesh point $\mathbf{x}_j, j = 1, \cdots, N$, and time step $t^n, n = 0, \cdots, M$, the unknowns are ordered in the order of $C_j^n, G_j^n, j = 1, \cdots, N, n = 0, \cdots, M$. Such ordering contains unknowns of the same space-time subdomain in a subblock, preconditioners such as additive Schwarz can be applied naturally to each subblock of the fully coupled KKT system and the ordering also improves the cache performance of the LU factorization based solvers. Since $\mathbf{f}$ is defined only in the time dimension, we put all the unknowns of $\mathbf{f}$ at the end after C and G. More precisely, we define the solution vector U by

$$U = (C_1^0, G_1^0, \cdots, C_N^0, G_N^0, C_1^1, G_1^1, \cdots, C_N^1, G_N^1, \cdots, C_1^M, G_1^M,$$
$$\cdots, C_N^M, G_N^M, f_1^0, \cdots, f_s^0, \cdots, f_1^M, \cdots, f_s^M)^T$$

then the linear system (13) with unknowns C_j^n and G_j^n, $j = 1, \cdots, N$, $n = 0, \cdots, M$, and f_k^n, $k = 1, \cdots, s$, $n = 0, \cdots, M$, is reformulated into the following linear system:

(15)
$$\mathbf{F}U = b,$$

where $\mathbf{F}$ is a sparse matrix of size $(M+1)(2N+s) \times (M+1)(2N+s)$ derived from the finite element discretization for KKT system (14) with the following block

structure:

$$\mathbf{F} = \begin{pmatrix} S_{00} & S_{01} & \mathbf{0} & \cdots & \mathbf{0} & S_{0,M+1} \\ S_{10} & S_{11} & S_{12} & \cdots & \mathbf{0} & S_{1,M+1} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & S_{M-1,M-2} & S_{M-1,M-1} & S_{M-1,M} & S_{M-1,M+1} \\ \mathbf{0} & \cdots & \mathbf{0} & S_{M,M-1} & S_{M,M} & S_{M,M+1} \\ S_{M+1,0} & S_{M+1,1} & S_{M+1,2} & \cdots & S_{M+1,M} & S_{M+1,M+1} \end{pmatrix},$$

and b has the following form correspondingly:

$$b = (b_0, b_1, \cdots, b_{M+1})^T.$$

In the matrix $\mathbf{F}$, the block matrices S_{ij}, with $0 \le i, j \le M$ are of size $2N \times 2N$ and are zero matrices except the ones in tridiagonal stripes $\{S_{i,i-1}\}, \{S_{i,i}\}, \{S_{i,i+1}\}$. The stripe $\{S_{i,M+1}\}$, $0 \le i \le M$ are nonzero sparse blocks of size $2N \times s(M+1)$; furthermore $\{S_{M+1,i}\}$, $0 \le i \le M$ are nonzero sparse blocks of size $s(M+1) \times 2N$ and $S_{M+1,M+1}$ is a nonzero tridiagonal matrix of size $s(M+1) \times s(M+1)$.

4.2. **Space-time Schwarz preconditioners.** The KKT system (15) is usually large in size and severely ill-conditioned. In traditional reduced space SQP methods, the subsystems corresponding to the unknowns C and G are time-dependent, time-marching algorithms starting from the initial or terminal moment are applied. But we notice that the adjoint equation needs the concentration distribution of C at the terminal time $t = T$, which means that the state equation and the adjoint equation should be solved in a sequential order. In addition, sequential steps within reduced space SQP methods exist between both the KKT subsystems and the time marching for time-dependent inverse problems, thus are quite challenging for efficient parallelization. To overcome the lack of parallelism in SQP methods, we shall propose to solve the fully coupled system (15) all at once. This is a very large system, the all-at-once method is traditionally regarded as a very expensive approach and not suitable for small computers. But on high-performance computers, especially on the upcoming exascale computers, we believe this approach is more attractive than the reduced space methods. It is well known that a direct solver such as Gaussian elimination or LU factorization is not suitable for very large problems due to the lack of parallel scalability. We shall use a preconditioned Krylov subspace method, where some preconditioning technique will be introduced for reducing the condition number of the KKT system and accelerating the convergence rate of the Krylov subspace method. Various preconditioners have been developed and applied for various elliptic and parabolic systems, such as the (block) Jacobi method, (incomplete) LU factorization, (multiplicative) additive Schwarz method, multigrid method, multilevel method, etc. [7, 8, 9]. Among these preconditioners the Schwarz type domain decomposition method is shown to have excellent preconditioning effect and parallel scalability [9, 31].

We shall propose a "space-time" Schwarz type preconditioner for the unsteady inverse problems. Different from the classical Schwarz type preconditioning technique which only decomposes the space domain, we want full parallelization in both space and time. The idea of space-time parallel algorithm comes from the parareal algorithm, proposed by Lions et al. in [25]. The parareal algorithm is an iterative

method which involves a coarse (coarse grid in the time dimension) solver for pre-diction and a fine (fine grid in the time dimension) solver for correction. An insight on the stability and convergence of the parareal algorithm was given in [16, 38]. Parareal algorithm has been applied to solve problems in molecular dynamics [4], fluid and structural mechanics [15], quantum control [28] etc. However in the implementation of the parareal algorithm, the scalability is determined largely by the coarse time step and the space discretization scheme. In [29], the parareal algorithm was combined with domain decomposition in space to achieve higher degree of parallelization. Different from the parareal algorithm, the new "space-time" Schwarz type preconditioner treats the time variable and the space variables equally, so the physical domain is a "space-time" domain, instead of the conventional space domain. We apply a domain decomposition technique to the coupled "space-time" domain.

In each "space-time" subdomain, a time-dependent subproblem with vanishing space boundary conditions and vanishing data at "artificial" initial and terminal time is solved. The same as the global problem, no time-marching is performed in each subproblem, all unknowns associated to the same space-time subdomain are solved simultaneously. The proposed "space-time" Schwarz preconditioner eliminates all sequential steps and all unknowns are treated at the same level of priority. We use a right-preconditioned restarted GMRES to solve the system (15):

$$\mathbf{F}M^{-1}U' = b,$$

where M^{-1} is a "space-time" additive Schwarz preconditioner and $U = M^{-1}U'$.

To formally define the preconditioner M^{-1} we need to introduce a partition of the space-time domain $\Omega \times [0, T]$, denoted by Θ. Firstly we decompose the domain Ω into nonoverlapping subdomains Ω_i, $i = 1, \cdots, N_1$, and then divide the time interval $[0, T]$ into subintervals $T_j = [t_{j-1}, t_j]$, $j = 1, 2, \cdots, N_2$, and $0 = t_0 < t_1 < \cdots < t_{N_2} = T$. We remark that the time partition here is coarser than that used in the full finite element discretization described in Section 3, and each interval T_j contains a few consequent time intervals $[t^k, t^{k+1}]$. Θ consists of $\Theta_{ij} = \Omega_i \times T_j$, $i = 1, \cdots, N_1, j = 1, \cdots, N_2$. In order to obtain an overlapping decomposition of Θ, we extend each subdomain Ω_i to a larger region Ω_i' and each subinterval T_j to a longer interval T_j', satisfying $\Omega_i \subset \Omega_i'$, $T_j \subset T_j'$. Now each Θ_{ij} can be straightforwardly extended to $\Theta_{ij}' = \Omega_i' \times T_j'$ with $\Theta_{ij} \subset \Theta_{ij}'$. The sizes of Θ_{ij}' are chosen so that the overlap is as uniform as possible around the perimeter of interior domains $\Theta_{ij}' \subset \Theta$. For boundary subdomains we neglect the part outside of Θ. See Figure 1 for an illustration of the space-time domain decomposition.

We denote the size of the KKT matrix $\mathbf{F}$ by $\tilde{N} \times \tilde{N}$, clearly there are two degrees of freedom at each mesh point corresponding to the state variable C and the adjoint variable G. The unknown time-dependent source intensity variables are allocated on the same processor as the last space-time subdomain Θ_{N_1,N_2}'.

On each extended subdomain Θ_{ij}', we define the $\tilde{N}_{ij} \times \tilde{N}$ matrix R_{ij}^δ, its 2×2 block element $(R_{ij}^\delta)_{l_1,l_2}$ is either an identity block if the integer indices l_1 and l_2 are related to the same mesh point and time step and they belong to Θ_{ij}' or a zero block otherwise. The multiplication of R_{ij}^δ with an $\tilde{N} \times 1$ vector generates a shorter vector by keeping all components corresponding to the subdomain Θ_{ij}'. R_{ij}^0 is defined similarly as R_{ij}^δ, with the difference that its application to a $\tilde{N} \times 1$ vector excludes the mesh points in $\Theta_{ij}' \backslash \Theta_{ij}$. Now for each space-time subdomain we have
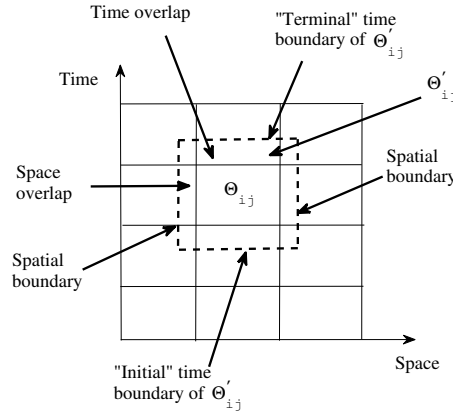
Figure 1. "Space-time" domain decomposition - an overlapping subdomain with boundary conditions.

defined the following local problem:

$$(16) \quad \begin{cases} \dfrac{\partial C}{\partial t} = \nabla \cdot (a(\mathbf{x})\nabla C) - \nabla \cdot (\mathbf{v}(\mathbf{x})C) + \sum_{i=1}^{s} \delta(\mathbf{x} - \mathbf{x}_i^*)f_i(t), \ (\mathbf{x}, t) \in \Theta'_{ij} \\ \dfrac{\partial G}{\partial t} = -\nabla \cdot (a(\mathbf{x})\nabla G) - \mathbf{v}(\mathbf{x}) \cdot \nabla G \\ \qquad + \delta(t - T)(C(\mathbf{x}, t) - C^{\varepsilon}(\mathbf{x}, t)), \ (\mathbf{x}, t) \in \Theta'_{ij}. \end{cases}$$

It is complemented by the following boundary conditions

$$(17) \qquad\qquad C(\mathbf{x}, t) = 0; \quad G(\mathbf{x}, t) = 0, \ \mathbf{x} \in \partial \Omega'_i$$

along with the "initial" and "terminal" time boundary conditions

$$(18) \qquad\qquad C(\mathbf{x}, t_{j-1}) = 0; \quad G(\mathbf{x}, t_{j-1}) = 0$$

$$(19) \qquad\qquad C(\mathbf{x}, t_j) = 0; \quad G(\mathbf{x}, t_j) = 0.$$

For the last subdomain, we include the additional variables corresponding to the source intensities f_i, $i = 1, \cdots, s$ satisfying

$$(20) \qquad\qquad \beta_2^i f_i'' + \beta_1^i f_i + G(\mathbf{x}^*, \cdot) = 0,$$

with the Neumann condition

$$(21) \qquad\qquad f_i'(t) = 0, \qquad t = 0, T.$$

We remark that (16) is a parabolic system and it is usually "illegal" to impose both the initial and terminal conditions (18)-(19). However, as inexact local solvers on space-time subdomains that form the global preconditioner, such local boundary conditions work well as we shall see from our numerical experiments. Similar boundary conditions are used in the context of hyperbolic subdomain problems [43].

Let M_{ij} be a discretization of (16)-(19) and M_{ij}^{-1} be an exact or approximate inverse of M_{ij}. The space-time additive Schwarz preconditioner is defined as

$$M_{asm}^{-1} = \sum_{j=1}^{N_2} \sum_{i=1}^{N_1} (R_{ij}^{\delta})^T M_{ij}^{-1} R_{ij}^{\delta}.$$

It is noted that the last space-time subdomain solver M_{N_1,N_2}^{-1} is an inverse or an approximate inverse of the matrix arising from the discretization of the subproblem (16)-(19) of Θ'_{N_1,N_2} and (20)-(21). Although its construction is slightly different from that of the other subdomain inverse matrices, we still use the same notation.

In addition to the standard additive Schwarz method (ASM) described above, the restricted version (RAS) of the method developed in [10] for standard space domain decompositions is also widely used. So we extend it to our current space-time domain decomposition, then the space-time RAS preconditioner is defined as

$$M_{ras}^{-1} = \sum_{j=1}^{N_2} \sum_{i=1}^{N_1} (R_{ij}^\delta)^T M_{ij}^{-1} R_{ij}^0.$$

For some applications, RAS achieves better preconditioning effect with less communication time since one of the restriction or extension operations does not involve any overlap. We use the restricted version in our experiments to be presented in the next section.

We remark that, computationally, the matrix M_{ij} can be obtained as $R_{ij}^\delta \mathbf{F}(R_{ij}^\delta)^T$. Moreover, if $N_2 = 1$, then no time partition is performed in the time dimension, M_{ras}^{-1} is a "space-only" domain decomposition preconditioner for the fully coupled KKT system.

5. **Numerical examples.** We present in this section some numerical examples of recovering the intensity functions $f_i(t)$ $(i = 1, \cdots, s)$ at given source locations $\mathbf{x}_1^*, \cdots, \mathbf{x}_s^*$. We set the test domain to be $\Omega = (-2, 2) \times (-2, 2)$, and the terminal time at $T = 1$. We denote the time step by n_t and the number of mesh points in x and y directions by n_x and n_y, respectively. Homogeneous Dirichlet and Neumann boundary conditions are imposed on $\Gamma_1 = \{\mathbf{x} = (x, y); |x| = 2\}$ and $\Gamma_2 = \{\mathbf{x} = (x, y); |y| = 2\}$, respectively. The diffusive coefficient $a(\mathbf{x})$ and the convective coefficient $\mathbf{v}(\mathbf{x})$ are chosen to be 1.0 and $(1.0, 1.0)^T$, respectively.

The preconditioned KKT coupled system will be solved by the restarted GMRES method with a restart number 50 [33]. For clarity, we provide the restarted GMRES algorithm below for a general linear system $Ax = b$:

Algorithm 1 Restarted GMRES method with a restart number m

---

   1. Compute $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$, and $v_1 = r_0/\beta$;
   2. Generate the Arnoldi basis and the matrix $\bar{H}_m$ using the Arnoldi algorithm starting with v_1;
   3. Compute y_m which minimizes $\|\beta e_1 - \bar{H}_m y\|_2$ and $x_m = x_0 + V_m y_m$;
   4. Stop the iteration if the stopping criteria are satisfied; otherwise set $x_0 := x_m$ and go to 1.

In the algorithm above, e_1 is the first column of the identity matrix, V_m stands for the $n \times m$ matrix with columns $v_1, \cdots, v_m$, and $\bar{H}_m$ denotes the $(m + 1) \times m$ Hessenberg matrix [33]. The computational cost is overwhelming and becomes more unstable numerically when m is large. So we should set m to a reasonable number and get restarted when the stopping criteria are not satisfied.

For definiteness, we shall denote as *a cell* the smallest space-time element after the space triangulation of Ω and time partition of the interval $[0, T]$. The size of overlap, that is the number of overlapping cells, is denoted by *iovlp* and set to 4

unless otherwise specified. The subsystem is solved with a sparse LU factorization or an incomplete LU factorization (ILU) with the fill-in level denoted by *ilulevel*. We still take the linear system $Ax = b$ for example to explain the definition of the fill-in level of an ILU factorization. Based on the Gaussian elimination, each location (i, j) of the sparse matrix A has a level of fill, denoted by lev_{ij}, which should indicate that the higher the level is, the smaller the element is [33]:

1. The initial value of fill of an element a_{ij} of A is defined by

$$\text{lev}_{ij} = \begin{cases} 0 & \text{if } a_{ij} \neq 0 \text{ or } i = j \\ \infty & \text{otherwise} \end{cases}$$

2. Since in the process of Gaussian elimination, the element a_{ij} is updated in the loop of k by $a_{ij} = a_{ij} - a_{kj}(a_{ik}/a_{kk})$, the corresponding level of fill is updated by $\text{lev}_{ij} = \min\{\text{lev}_{ij}, \text{lev}_{ik} + \text{lev}_{kj} + 1\}$.

In ILU with fill-in level p, an element whose level of fill lev_{ij} does not exceed p will be kept. So the larger the level of fill p is, the more elements are kept in the factorization.

For the scalability test we use LU factorization as the subdomain solver and incomplete LU factorization with *ilulevel* $= 3$ for the other tests if not specified. The relative residual convergence tolerance of GMRES is set to be 10^{-5}. The algorithm is implemented based on the package Portable, Extensible Toolkit for Scientific computation (PETSc) [6] and run on a Dawning TC3600 blade server system at the National Supercomputing Center in Shenzhen, China with a 1.271 PFlops/s peak performance.

We use a high resolution numerical solution of the concentration at the terminal time $t = T$ as the noise-free observation data. In other words, we first solve the forward convection-diffusion system (1)-(3) on a very fine mesh, 640×640, with a small time stepsize $\tau = 1/160$, then add a random noise of the following form to the terminal concentration

$$C^\varepsilon(\mathbf{x}) = (1 + \varepsilon\, r)C(\mathbf{x}, T),$$

where r is a random function with uniform distribution in $[-1, 1]$, and ε is the noise level. In our numerical experiments, ε is set to 1% if not specified.

5.1. **Reconstruction results and parallel efficiency tests.** The tests are designed to investigate the recovery effect of the pollutant source intensity functions and to understand how the solution of the KKT system behaves when using different mesh sizes, time steps, regularization parameters and number of processors, which is denoted by *np*. Supposing the source locations are known, we consider the following four examples.

(1) $f = t^2,$ $\qquad\qquad\qquad\qquad$ $\mathbf{x}_1^* = (1.0, 1.0)^T.$

(2) $f = \dfrac{75}{4}t(1-t)\left(\dfrac{1}{6} - t\right)^2 + 1.0,$ $\qquad$ $\mathbf{x}_1^* = (1.0, 1.0)^T.$

(3) $f_1 = t^2,$ $\qquad\qquad\qquad\qquad$ $\mathbf{x}_1^* = (1.0, -1.0)^T$

$\qquad f_2 = \dfrac{75}{4}t(1-t)\left(\dfrac{1}{6} - t\right)^2 + 1.0,$ $\qquad$ $\mathbf{x}_2^* = (0.0, 0.0)^T.$

(4) $f_1 = t^2,$ $\qquad\qquad\qquad\qquad$ $\mathbf{x}_1^* = (34/79, 24/79)^T$

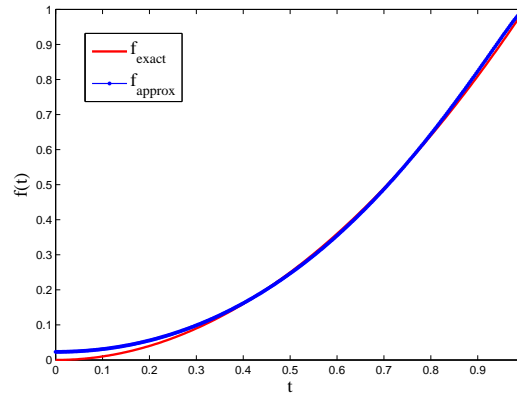$\qquad f_2 = \dfrac{75}{4}t(1-t)\left(\dfrac{1}{6} - t\right)^2 + 1.0,$ $\qquad$ $\mathbf{x}_2^* = (14/79, 14/79)^T$

FIGURE 2. Comparison of analytical and computed solution for Example 1.

np	Its	Time(sec)	Speedup	Ideal
256	145	794.70	1	1
512	181	377.16	2.11	2
1024	242	161.23	4.93	4

TABLE 1. Scalability test for Example 1: $n_t = 160$, $n_x = 160$, $n_y = 160$, $DOF = 8,192,160$.

$$f_3 = 3 - t, \qquad\qquad \mathbf{x}_3^* = (25/79, 15/79)^T.$$

Example 1. This is an example of recovering a quadratic polynomial source intensity function. An H^1 regularization is applied and the parameter is chosen to be $\beta_2 = 10^{-4}$ ($\beta_1 = 0$). Figure 2 shows the reconstructed result with mesh $n_x = 80, n_y = 80$ and time step $n_t = 320$, when 64 processors are used. The blue dotted line represents the reconstructed source intensity which is quite close to the red true shape. This shows that the time-dependent intensity is successfully recovered by the algorithm.

We present the strong scalability results in Table 1. Sparse LU factorization is applied as the subdomain solver. The spatial mesh is 160×160 and the number of time steps is 160. The total degrees of freedom is $8,192,160$. As the number of processors increases, the computing time decreases significantly and superlinear speedup is obtained, for $np \leq 1024$, in Figure 3. Since the number of processors is the same as the number of subdomains, more processors lead to an increasing number of iterations. This suggests that the condition number of the preconditioned KKT matrix depends on the number of subdomains. Similar dependency was proved for elliptic problems [39].

We fix the number of processors to $np = 128$, the mesh to $n_x = 80$, $n_y = 80$ and the time step $n_t = 320$, then test several choices of regularization parameters. ILU factorization is used as the subdomain solver with the fill-in level being $ilulevel = 3$. From the results in Table 2, as β_2 becomes smaller, the number of GMRES iterations increases, and no significant change is observed for the total computing time.
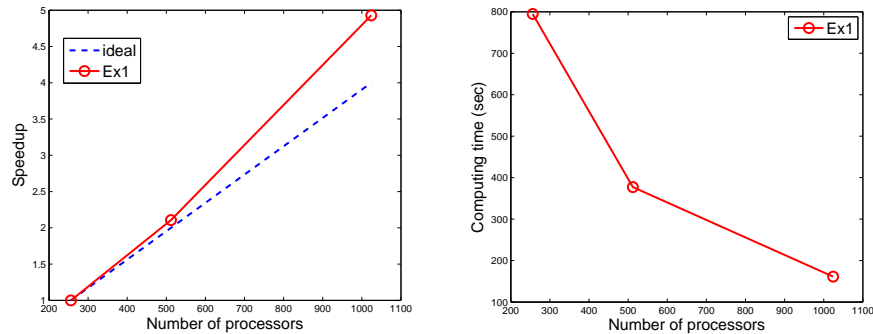
Figure 3. The speedup (left) and computing time (right) for Example 1.

β_2	Its	Time(sec)
10^{-4}	88	54.91
10^{-5}	93	55.75
10^{-6}	96	56.68

Table 2. H^1 regularization parameter test for Example 1: $n_t = 320, n_x = 80, n_y = 80, DOF = 4,096,320, np = 128$.

Example 2. This is an example of recovering a polynomial source intensity function of degree 4. We set $\beta_1 = 0$ and use an H^1 regularization with $\beta_2 = 10^{-4}$. Satisfactory result is shown in Figure 4 with mesh $n_x = 80, n_y = 80$ and time step $n_t = 160$, when 64 processors are used for the computation.

Using the same parameter settings as in Example 1, we perform the strong scalability test and the results are given in Table 3 and Figure 5. Superlinear speedup is obtained when $np \leq 1024$. Next we test three sets of mesh and time step size in Table 4. The H^1 regularization parameter is set to be $\beta_2 = 10^{-6}$, and 64 processors are used. The overlap $iovlp = 4$ and the fill-in level of ILU $ilulevel = 3$. We observe from Table 4 that as the mesh and the time step size become finer, the number of GMRES iterations grows slightly, and the computing time increases with the problem size.

Now we investigate the performance of the space-time Schwarz preconditioner. An important feature of the proposed space-time Schwarz preconditioner lies in the parallelization in the time dimension. If the time range is not partitioned as mentioned in the end of Section 4.2, the preconditioner also works from the result in Figure 6, but it is observed from Table 5, under the same settings, that the "space-only" Schwarz preconditioner costs more iterations and computing time compared to the space-time Schwarz preconditioner. Thus the Schwarz preconditioner with a partition in the time is more efficient than the "space-only" domain decomposition preconditioner. In the end of this example, we perform the noise level test and the results are given in Figure 7. The results agree with our expectation that the reconstruction accuracy deteriorates with the increasing level of noise in the measurement data.

Example 3. In this example, we test the recovery of two source intensities. Compared with the single source intensity cases, more regularization parameters are
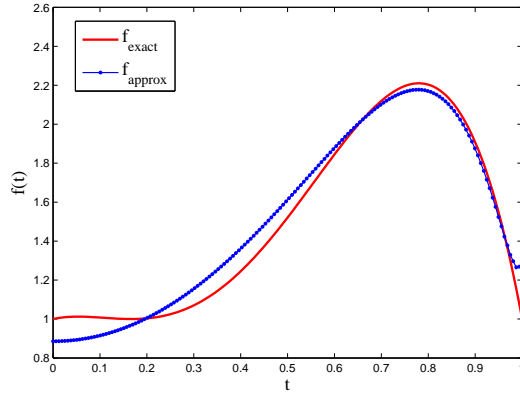
FIGURE 4. Comparison of analytical and computed solution for Example 2.

np	Its	Time(sec)	Speedup	Ideal
256	178	852.23	1	1
512	184	379.75	2.24	2
1024	247	176.23	4.84	4

TABLE 3. Scalability test for Example 2: $n_t = 160$, $n_x = 160$, $n_y = 160$, $DOF = 8,192,160$.
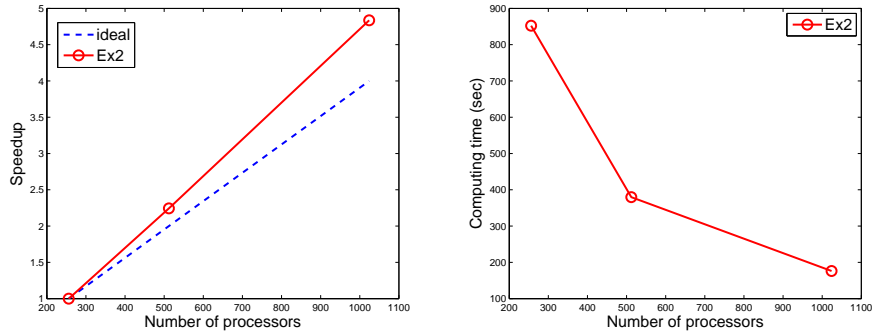

FIGURE 5. The speedup (left) and computing time (right) for Example 2.

n_t	$n_x \times n_y$	DOF	Its	Time(sec)
100	40×40	320 100	40	8.44
240	64×64	1 966 320	46	33.25
320	80×80	4 096 320	49	69.28

TABLE 4. Mesh size and time step size test for Example 2: $\beta_2 = 10^{-6}$, $np = 64$.

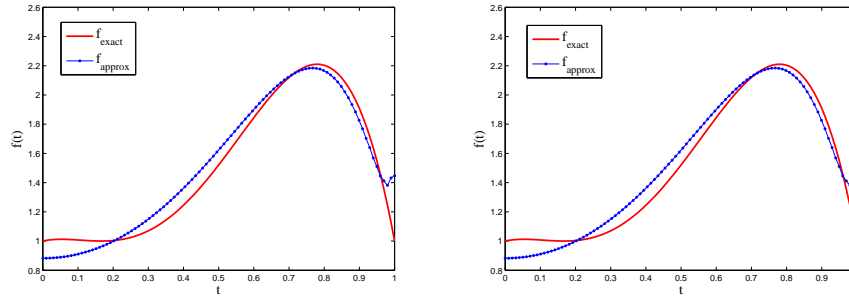FIGURE 6. Space-only ASM preconditioner (left) vs. space-time ASM preconditioner (right) for Example 2.

Preconditioner	np	Its	Time(sec)	np	Its	Time(sec)
space-only	64	49	114.34	256	129	230.29
space-time	64	39	37.60	256	83	156.64

TABLE 5. Preconditioner comparison for Example 2: $\beta_2 = 10^{-5}$, $n_t = 100$, $n_x = 40$, $n_y = 40$, $DOF = 320,100$ for $np = 64$; $\beta_2 = 10^{-6}$, $n_t = 320$, $n_x = 80$, $n_y = 80$, $DOF = 4,096,320$ for $np = 256$.
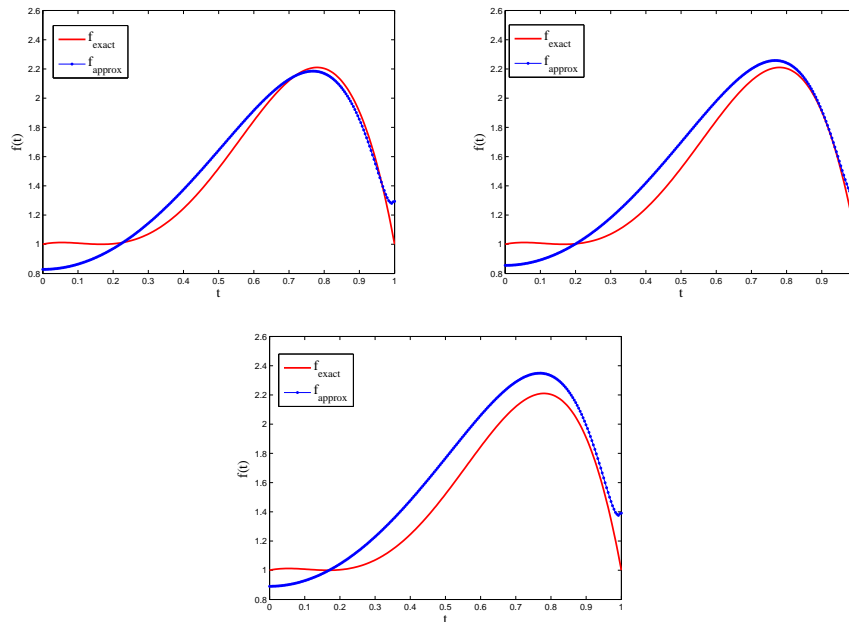

FIGURE 7. Noise level test for example 2. Top left: $\varepsilon = 1\%$, top right: $\varepsilon = 5\%$, bottom: $\varepsilon = 10\%$. $n_t = 240$, $n_x = 64$, $n_y = 64$, $DOF = 1,966,320$, $\beta_2 = 10^{-5}$, $np = 64$.
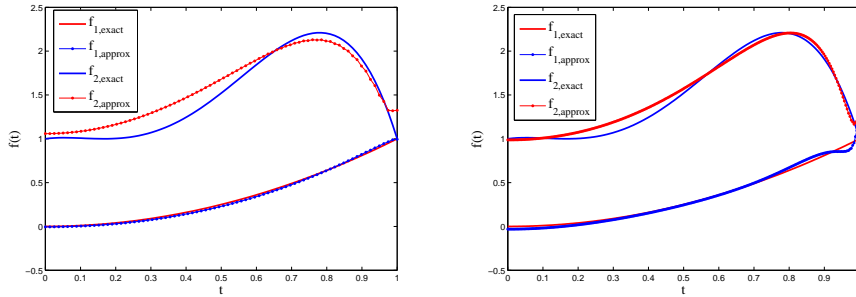
FIGURE 8. Recovery of two source intensities using two sets of regularization parameters. Left: $\beta_1^1 = 0, \beta_1^2 = 10^{-4}, \beta_2^1 = 10^{-6}, \beta_2^2 = 10^{-5}$, right: $\beta_1^1 = 0, \beta_1^2 = 10^{-6}, \beta_2^1 = 0, \beta_2^2 = 10^{-6}$.

np	Its	Time(sec)	Speedup	Ideal
256	148	765.80	1	1
512	150	360.88	2.12	2
1024	211	178.37	4.29	4

TABLE 6. Scalability test for Example 3 with two point sources: $n_t = 160$, $n_x = 160$, $n_y = 160$, $DOF = 8,192,320$.

needed. Here we test the following two sets of regularization parameters with $np = 64$:

(1) $\beta_1^1 = 0, \beta_1^2 = 10^{-4}$ for the source intensity function f_1 and $\beta_2^1 = 10^{-6}, \beta_2^2 = 10^{-5}$ for f_2. The mesh is $n_x = 80, n_y = 80$ and the time step is $n_t = 80$;
(2) $\beta_1^1 = 0, \beta_1^2 = 10^{-6}$ for f_1 and $\beta_2^1 = 0, \beta_2^2 = 10^{-6}$ for f_2. The mesh and the time step are set to be $n_x = 64, n_y = 64$ and $n_t = 256$.

The numerical results are shown in Figure 8. The computed f_1 matches with its original data perfectly, but the computed f_2 is less accurate. As we see in the tests for Example 1 and Example 2, f_2 is physically harder to recover than the simpler function f_1. Overall the reconstruction effect for both source intensities are reasonable.

Next we show the strong scalability results in Figure 9 and Table 6. We still observe a superlinear speedup, although it is a bit worse than that of Examples 1 and 2. It implies that it is more difficult to separate and identify multiple source intensities than the single source case. And from our previous experiments with reduced space SQP methods, the recovery of multiple sources is much more difficult to converge than that of the single source case.

Lastly, we test the algorithm with parameters such as the fill-in level of ILU factorization and the size of overlap in Table 7 and Table 8, respectively.

It is observed that the number of iterations decreases with the increase of the overlapping size or the fill-in level, however, it costs more communication time when we increase the overlap between "space-time" subdomains, and more computing time is used in the preconditioning stage when we raise the fill-in level of the ILU factorization.

Example 4. We now test the numerical reconstruction for three point sources, and observe how the speedup changes with increasing number of sources.
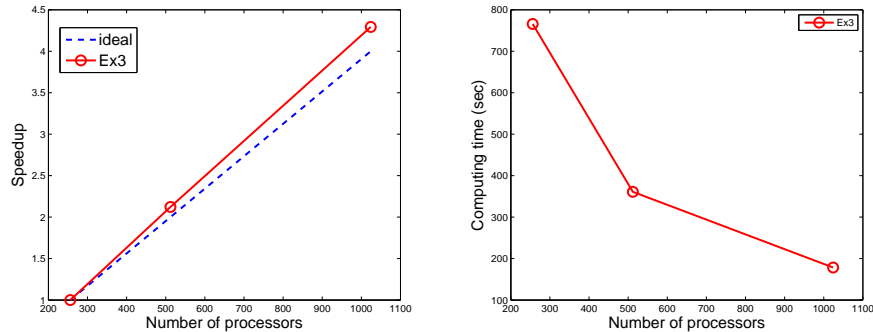
FIGURE 9. The speedup (left) and the computing time (right) for Example 3.

$ilulevel$	Its	Time(sec)
1	496	75.98
2	295	85.12
3	247	116.11

TABLE 7. Fill-in level of ILU test for Example 3: $\beta_1^2 = 10^{-6}$, $\beta_2^2 = 10^{-6}$, $n_t = 400$, $n_x = 80$, $n_y = 80$, $DOF = 5,120,400$, $np = 128$.

$iovlp$	Its	Time(sec)
1	-	-
2	432	114.22
4	247	121.26
6	238	400.12

TABLE 8. Overlap test for Example 3: $\beta_1^2 = 10^{-6}$, $\beta_2^2 = 10^{-6}$, $n_t = 400$, $n_x = 80$, $n_y = 80$, $DOF = 5,120,400$, $np = 128$.

For this test, we take the spatial mesh 160×160 and the number of time steps 160. Regularization parameters are respectively set to be $\beta_1^1 = 0, \beta_2^1 = 10^{-5}$, $\beta_1^2 = 10^{-4}, \beta_2^2 = 10^{-5}$ and $\beta_1^3 = 10^{-7}, \beta_2^3 = 8 \times 10^{-6}$. From Figure 10 we see that, apart from the initial part of the third intensity which is not quite close to the true values, the rest are recovered satisfactorily. Now we use a 160×160 space mesh and 160 time step to test the strong scalability and compute time in Figure 11 and Table 9. LU factorization is used as the subdomain solver. It is observed that the speedup for three point sources is almost linear, still satisfactory but a bit worse than Examples 1,2 and 3. As a conclusion the speedup deteriorates slowly with the number of unknown point sources.

5.2. **Comparisons with two reduced space SQP methods.** A reduced space method for reconstruction of the location and intensity of a single point pollutant source was developed in [13]. With the source location known in our current case, the process of reconstructing the source intensity described in [13] can be stated as follows:
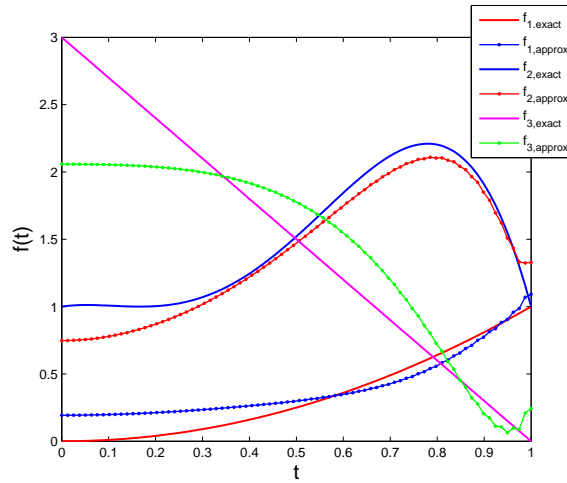
FIGURE 10. The reconstruction for three point sources.

np	Its	Time(sec)	Speedup	Ideal
256	148	794.19	1	1
512	150	383.56	2.07	2
1024	211	194.14	4.09	4

TABLE 9. Scalability test for Example 4 with three point sources: $n_t = 160$, $n_x = 160$, $n_y = 160$, $DOF = 8,192,480$.
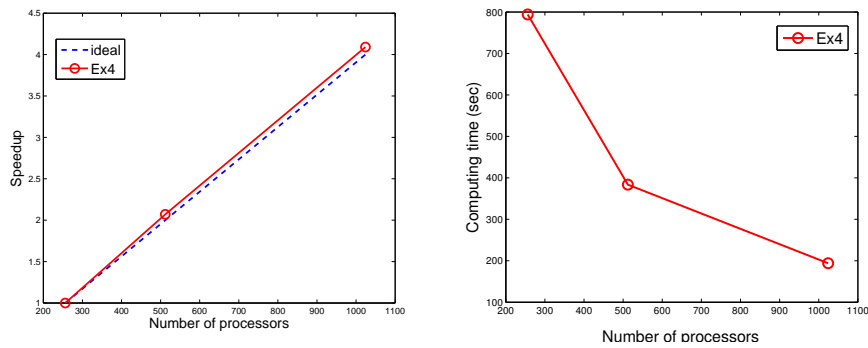

FIGURE 11. The speedup (left) and the computing time (right) for Example 4.

Algorithm 2 Nonlinear CG method

Select the initial guesses f^0, and set $k := 0$.
For $k = 1, 2, \cdots, N_{max}$
  Solve the state system (the first equation in (13)) for $\{C_h^n(f^k)\}$;
  Solve the adjoint system (the second equation in (13)) for $\{G_h^n(f^k)\}$;
  Apply the nonlinear CG method to update f^k: $f^{k+1} = f^k + \alpha_1^k d^k$;
  Stop the iteration if the stopping criteria are satisfied; otherwise set $k := k+1$.

We use the Fletcher-Reeves (FR) formula to update the nonlinear CG direction d^k: $d^k = J'_k + \gamma_k d^{k-1}$, with $d^0 = J'_0$ and $\gamma_k = \|J'_k\|^2/\|J'_{k-1}\|^2$ and $J'_k = -(J^\tau_h)'(f^k)$ being the negative gradient direction which is obtained from formula (9). That is,

$$J'_k = -\left(\int_0^T (\beta_1 f^k(t) - G^\tau_h(\mathbf{x}^*, t))g^\tau(t)dt + \beta_2 \int_0^T (f^k)'(t)(g^\tau)'(t)dt \right)$$

We select the stepsize α_1^k such that $\alpha_1^k = \text{argmin}_{\gamma>0} J^\tau_h(f^k + \gamma d_k)$. For the L^2 and H^1 regularizations in (5), we can work out the exact formulae:

$$\alpha_1^k = -\frac{(C^M_h(f^k) - C^\epsilon, A^M_h) + \beta_1(f^k, d^k)}{(A^M_h, A^M_h) + \beta_1(d^k, d^k)} \ (L^2 \text{ regularization}),$$

$$\alpha_1^k = -\frac{(C^M_h(f^k) - C^\epsilon, A^M_h) + \beta_2((f^k)', (d^k)')}{(A^M_h, A^M_h) + \beta_2((d^k)', (d^k)')} \ (H^1 \text{ regularization}),$$

where $A^M_h = C^M_h(f^k)'d^k$ is obtained by solving the following sensitivity equation,

$$(\partial_\tau A^n_h, v_h) + (a\nabla \bar{A}^n_h, \nabla v_h) + (\nabla \cdot (\mathbf{v}\bar{A}^n_h), v_h) = v_h(\mathbf{x}^*)(\bar{d}^k)^n, \ \forall v_h \in \mathring{V}^h, \ n = 1, \cdots, M,$$

with $A^0_h = \mathbf{0}$. At each iteration, three time-dependent subsystems are solved. When we implement this nonlinear CG method on parallel computers, we need to develop a parallel solver for each subsystem. We will test two cases: the first one uses the space domain decomposition preconditioner but keeps the time marching process, while the second one uses a space-time domain decomposition preconditioner as it is developed for the fully coupled system in this work and solves each time-dependent subsystem all-at-once. These two parallel solvers are denoted by RS(1) and RS(2) respectively. We shall compare the computing time between our proposed space-time preconditioning method, denoted by FS, and the two reduced space SQP methods RS(1) and RS(2); see Table 10. We use the three aforementioned methods to implement Example 2 with four sets of meshes, and the number of processors increases with the refinement of the meshes. The subdomain solvers for all three kinds of methods are ILU. We firstly compute the result by the FS method with zero initial guess and record the error accuracy $e = \|f - f^*\|$. Then we use the same initial guess and the error bound e for the reduced space methods RS(1) and RS(2), and set the stopping criterium as $\|f^k - f^*\| < e$. In this way we can compare the computing time for all these methods.

As shown in Table 10, the computing time of the FS method is much less than the ones of RS(1) and RS(2). For the two reduced space methods, RS(2) using the space-time domain decomposition solver is faster than RS(1) keeping the time marching process and using a space domain decomposition solver. We can see that the space-time fully coupled preconditioner is much better for parallellization, and the all-at-once method for the fully coupled KKT system is always more efficient than the reduced space iterative optimization method on parallel systems.

6. **Concluding remarks.** We developed a new space-time domain decomposition method for unsteady source inversion problems. The main ingredient of our algorithm includes solving the fully coupled KKT system by GMRES iteration with a space-time additive Schwarz preconditioner. Although the size of the linear system is significantly increased compared to the reduced space SQP methods, the one-shot method avoids the sequential step between the state equation and the adjoint equation, as well as the time-marching process in the time dimension, and thus achieves higher degree of parallelism. This is well confirmed by the numerical results shown

np	n_t	$n_x \times n_y$	Solver	Time(sec)
64	40	40×40	FS	12.064
			RS(1)	418.580
			RS(2)	125.484
128	80	80×80	FS	15.525
			RS(1)	682.794
			RS(2)	99.528
256	160	80×80	FS	23.736
			RS(1)	994.962
			RS(2)	200.543
512	320	160×160	FS	136.717
			RS(1)	7240.881
			RS(2)	1094.886

TABLE 10. The computing time of the proposed full space method FS, the reduced space method RS(1) and RS(2).

in the last section. Another advantage of the new method is that the recovery of multiple sources is obtained using the same algorithmic and software framework as the single source case, and the framework is easily extended to recover other kinds of source intensities.

We have observed from the numerical examples that the new space-time additive Schwarz method is quite robust also with respect to the noise in the observation data. It is important to note that the new space-time method is highly parallel and scalable, and extensible naturally to three-dimensional problems.

REFERENCES

[1] V. Akcelik, G. Biros, A. Draganescu, O. Ghattas, J. Hill and B. Waanders, Dynamic data-driven inversion for terascale simulations: Real-time identification of airborne contaminants, *Proceedings of Supercomputing 2005*, Seattle, WA, 2005, p43.

[2] V. Akcelik, G. Biros, O. Ghattas, K. R. Long and B. Waanders, A variational finite element method for source inversion for convective-diffusive transport, *Finite Elements in Analysis and Design*, **39** (2003), 683–705.

[3] J. Atmadja and A. C. Bagtzoglou, State of the art report on mathematical methods for groundwater pollution source identification, *Environmental Forensics*, **2** (2001), 205–214.

[4] L. Baflico, S. Bernard, Y. Maday, G. Turinici and G. Zerah, Parallel-in-time molecular-dynamics simulations, *Physical Review E*, **66** (2002), 2–5.

[5] V. Balakrishnan, All about the Dirac Delta function(?), *Resonance*, **8** (2003), 48–58.

[6] S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith and H. Zhang, *PETSc Users Manual*, Technical report, Argonne National Laboratory, 2014.

[7] A. Battermann, *Preconditioners for Karush-Kuhn-Tucker Systems Arising in Optimal Control*, Master Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1996.

[8] G. Biros and O. Ghattas, Parallel preconditioners for KKT systems arising in optimal control of viscous incompressible flows, in *Parallel Computational Fluid Dynamics 1999,* Towards Teraflops, Optimization and Novel Formulations, 2000, 131–138.

[9] X.-C. Cai, S. Liu and J. Zou, Parallel overlapping domain decomposition methods for coupled inverse elliptic problems, *Communications in Applied Mathematics and Computational Science*, **4** (2009), 1–26.

[10] X.-C. Cai and M. Sarkis, A restricted additive Schwarz preconditioner for general sparse linear systems, *SIAM Journal on Scientific Computing*, **21** (1999), 792–797.

[11] Z. Chen and J. Zou, An augmented Lagrangian method for identifying discontinuous parameters in elliptic systems, *SIAM Journal on Control and Optimization*, **37** (1999), 892–910.

[12] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, 1st edition, North-Holland Pub. Co., Amsterdam/New York, 1978.

[13] X. M. Deng, Y. B. Zhao and J. Zou, On linear finite elements for simultaneously recovering source location and intensity, *International Journal of Numerical Analysis and Modeling*, **10** (2013), 588–602.

[14] H. W. Engl, M. Hanke and A. Neubauer, *Regularization of Inverse Problems*, Mathematics and its Applications, 375, Kluwer Academic Publishers Group, Dordrecht, 1996.

[15] C. Farhat and M. Chandesris, Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications, *International Journal for Numerical Methods in Engineering*, **58** (2003), 1397–1434.

[16] M. J. Gander and E. Hairer, Nonlinear convergence analysis for the parareal algorithm, *Domain Decomposition Methods in Science and Engineering XVII*, Lect. Notes Comput. Sci. Eng., 60, Springer, Berlin, 2008, 45–56.

[17] S. Gorelick, B. Evans and I. Remson, Identifying sources of groundwater pullution: An optimization approach, *Water Resources Research*, **19** (1983), 779–790.

[18] E. Haber, A parallel method for large scale time domain electromagnetic inverse problems, *Applied Numerical Mathematics*, **58** (2008), 422–434.

[19] A. Hamdi, The recovery of a time-dependent point source in a linear transport equation: Application to surface water pollution, *Inverse Problems*, **25** (2009), 075006, 18pp.

[20] A. Hamdi, Identification of a time-varying point source in a system of two coupled linear diffusion-advection- reaction equations: Application to surface water pollution, *Inverse Problems*, **25** (2009), 115009, 21pp.

[21] B. Jin and J. Zou, Numerical estimation of the Robin coefficient in a stationary diffusion equation, *IMA Journal of Numerical Analysis*, **30** (2010), 677–701.

[22] Y. L. Keung and J. Zou, Numerical identifications of parameters in parabolic systems, *Inverse Problems*, **14** (1998), 83–100.

[23] Y. L. Keung and J. Zou, An efficient linear solver for nonlinear parameter identification problems, *SIAM Journal on Scientific Computing*, **22** (2000), 1511–1526.

[24] H. W. Kuhn and A. W. Tucker, Nonlinear programming, in *Proceedings of 2nd Berkeley Symposium*, University of California Press, Berkeley, 1951, 481–492.

[25] J.-L. Lions, Y. Maday and G. Turinici, A "parareal" in time discretization of PDEs, *Comptes-Rendus de l'Academie des Sciences Series I Mathematics*, **332** (2001), 661–668.

[26] X. Liu, *Identification of Indoor Airborne Contaminant Sources with Probability-Based Inverse Modeling Methods*, Ph.D. Thesis, University of Colorado, 2008.

[27] X. Liu and Z. Zhai, Inverse modeling methods for indoor airborne pollutant tracking literature review and fundamentals, *Indoor Air*, **17** (2007), 419–438.

[28] Y. Maday and G. Turinici, Parallel in time algorithms for quantum control: Parareal time discretization scheme, *International Journal of Quantum Chemistry*, **93** (2003), 223–228.

[29] Y. Maday and G. Turinici, The parareal in time iterative solver: A further direction to parallel implementation, in *Domain Decomposition Methods in Science and Engineering*, Lect. Notes Comput. Sci. Eng., 40, Springer, Berlin, 2005, 441–448.

[30] G. Nunnari, A. Nucifora and C. Randieri, The application of neural techniques to the modelling of time-series of atmospheric pollution data, *Ecological Modelling*, **111** (1998), 187–205.

[31] E. Prudencio, R. Byrd and X.-C. Cai, Parallel full space SQP Lagrange-Newton-Krylov-Schwarz algorithms for PDE-constrained optimization problems, *SIAM Journal on Scientific Computing*, **27** (2006), 1305–1328.

[32] R. Revelli and L. Ridolfi, Nonlinear convection-dispersion models with a localized pollutant source II–a class of inverse problems, *Mathematical and Computer Modelling*, **42** (2005), 601–612.

[33] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd edition, Society for Industrial and Applied Mathematics, 2003.

[34] A. Samarskii and P. Vabishchevich, *Numerical Methods for Solving Inverse Problems of Mathematical Physics*, Walter de Gruyter, Berlin, 2007.

[35] T. Skaggs and Z. Kabala, Recovering the release history of a groundwater contaminant, *Water Resources Research*, **30** (1994), 71–79.

[36] T. Skaggs and Z. Kabala, Recovering the history of a groundwater contaminant plume: Method of quasi-reversibility, *Water Resources Research*, **31** (1995), 2669–2673.

[37] M. Snodgrass and P. Kitanidis, A geostatistical approach to contaminant source identification, *Water Resources Research*, **33** (1997), 537–546.

[38] G. Staff and E. Ronquist, Stability of the parareal algorithm, in *Domain Decomposition Methods in Science and Engineering*, Lect. Notes Comput. Sci. Eng., 40, Springer, Berlin, 2005, 449–456.

[39] A. Toselli and O. Widlund, *Domain Decomposition Methods-Algorithms and Theory*, Springer, 2005.

[40] J. Wong and P. Yuan, A FE-based algorithm for the inverse natural convection problem, *International Journal for Numerical Methods in Fluids*, **68** (2012), 48–82.

[41] A. Woodbury, *Inverse Engineering Handbook*, CRC Press, 2003.

[42] J. L. Xie and J. Zou, Numerical reconstruction of heat fluxes, *SIAM Journal on Numerical Analysis*, **43** (2005), 1504–1535.

[43] C. Yang, J. Cao and X.-C. Cai, A fully implicit domain decomposition algorithm for shallow water equations on the cubed-sphere, *SIAM Journal on Scientific Computing*, **32** (2010), 418–438.

[44] H. Yang, E. Prudencio and X.-C. Cai, Fully implicit Lagrange-Newton-Krylov-Schwarz algorithms for boundary control of unsteady incompressible flows, *International Journal for Numerical Methods in Engineering*, **91** (2012), 644–665.

[45] X. Zhang, C. X. Zhu, G. D. Feng, H. H. Zhu and P. Guo, Potential use of bacteroidales specific 16S rRNA in tracking the rural pond-drinking water pollution, *Journal of Agro-Environment Science*, **30** (2011), 1880–1887.

[46] B. Q. Zhu, Y. W. Chen and J. H. Peng, Lead isotope geochemistry of the urban environment in the Pearl River Delta, *Applied Geochemistry*, **16** (2001), 409–417.

E-mail address: xm.deng@siat.ac.cn

E-mail address: cai@colorado.edu

E-mail address: zou@math.cuhk.edu.hk