

A Variational Finite Element Method for Source Inversion for Convective-Diffusive Transport^{*}

Volkan Akçelik^a, George Biros^b, Omar Ghattas^a, Kevin R. Long^c,
Bart van Bloemen Waanders^d

^a*Mechanics, Algorithms, and Computing Laboratory
Departments of Biomedical and Civil & Environmental Engineering
Carnegie Mellon University, Pittsburgh, PA 15213, USA.*

^b*Courant Institute for Mathematical Sciences, New York University
New York, NY 10012, USA*

^c*Sandia National Laboratories, Livermore, CA 94551, USA*

^d*Sandia National Laboratories, Albuquerque, NM 87185, USA*

Abstract

We consider the inverse problem of determining an arbitrary source in a time-dependent convective-diffusive transport equation, given a velocity field and pointwise measurements of the concentration. Applications that give rise to such problems include determination of groundwater or airborne pollutant sources from measurements of concentrations, and identification of sources of chemical or biological attacks. To address ill-posedness of the problem, we employ Tikhonov and total variation regularization. We present a variational formulation of the first order optimality system, which includes the initial-boundary value state problem, the final-boundary value adjoint problem, and the space-time boundary value source problem. We discretize in the space-time volume using Galerkin finite elements. Several examples demonstrate the influence of the density of the sensor array, the effectiveness of total variation regularization for discontinuous sources, the invertibility of the source as the transport becomes increasingly convection-dominated, the ability of the space-time inversion formulation to track moving sources, and the optimal convergence rate of the finite element approximation.

^{*} Research supported by the Computer Science Research Institute (CSRI) at Sandia National Laboratories, by the National Science Foundation's Information Technology Research (ITR) program through grant ACI-0121667, and by the Department of Energy's Scientific Discovery Through Advanced Computing (SciDAC) program through the Terascale Optimal PDE Simulations (TOPS) Center.

Email addresses: volkan@andrew.cmu.edu (Volkan Akçelik),
biros@cs.nyu.edu (George Biros), oghattas@cs.cmu.edu (Omar Ghattas),

1 Introduction

Suppose we are given observations of the concentration field of an airborne or water-borne contaminant, along with a velocity field that convects it and a parameter that characterizes its diffusion. We are interested in determining the distribution and strength of the source of the contaminant. In this article, we consider such an inverse problem: namely, we wish to determine an arbitrary source function that drives a time-dependent convective-diffusive transport process, given a velocity field and observations of a concentration field at a finite number of points throughout a domain of interest.

Inverse problems of this type are typically ill-posed; the sensors providing the observations are usually coarsely spaced, and we cannot hope to recover components of the source function that are more oscillatory than dictated by the spacing of the sensors. A similar idea holds for sampling in time. Therefore, arbitrary oscillatory components will appear as noise in the reconstructed source function; although unobservable by the sensors, these oscillatory components add noise to the predicted concentration field. To address this source of ill-posedness, we employ Tikhonov regularization, which has the effect of filtering the most oscillatory components of the source. However, Tikhonov regularization also smoothes discontinuities, so for sources that have sharp fronts, we instead use total variation regularization, which preserves the discontinuities at the expense of giving rise to strong nonlinearities in the regularization term.

We formulate the transport source inversion problem as an infinite dimensional optimization problem that is constrained by the time-dependent convection-diffusion equation. The objective function is the squared discrepancy between observed and predicted concentrations at the sensor locations over time. The optimization variable is the source function in space and time. Optimality conditions for the inverse problem are derived in weak and strong form via calculus of variations. The system of optimality conditions consists of the initial-boundary value transport problem, the final-boundary value adjoint transport problem, and the space-time boundary value source problem. The weak form leads naturally to a Galerkin space-time finite element method that discretizes the space-time volume. This system can be rather formidable to solve. To implement the numerical solution, we leverage the high-level finite element symbolic toolkit *Sundance*, through which all of the meshing, discretization, element computation, assembly, and solution methods are abstracted away from the user. We present several numerical examples that examine the influence of the density of the sensor array on the quality of the reconstructed source; the effectiveness of total variation regularization for discontinuous sources; the invertibility of the source as the transport becomes increasingly convection-dominated;

krlong@ca.sandia.gov (Kevin R. Long), bartv@sandia.gov (Bart van Bloemen Waanders).

the ability of the space-time inversion formulation to track moving sources, and the rate of convergence of the finite element approximation of the reconstructed source to the exact solution of the inverse problem.

Source inversion for convective-diffusive transport has a number of environmental, energy, and national security applications. Examples include localizing the sources of chemical and biological attacks on public facilities (such as subways or airports), and determining air or groundwater pollutant sources. A first step towards solving problems of this type is the development of an appropriate formulation of the conditions that characterize optimality, as well as an associated numerical approximation of these conditions; this is the focus of this article. Scalable, fast, robust algebraic solvers for the optimality conditions are certainly of paramount interest, particularly for (near) real-time applications. The subject of solvers is largely beyond our present scope; we comment briefly on some of the issues and provide references to our related work, but defer a full treatment to future articles.

The literature on inverse transport problems is concerned mostly with radiation transport inverse problems, which are characterized by convective transport. This is not surprising, given the importance of applications in computerized tomography (see for example the survey by Natterer [10]). Recently, interest has begun to emerge in source inversion for atmospheric transport models, specifically estimating CO₂ surface fluxes from atmospheric observations [6]. While this problem class is somewhat different from those we consider (in particular unknown boundary fluxes, rather than a volume source, are to be estimated), there are nevertheless many similarities, and the method described here may provide useful for such problems (and vice versa).

The remainder of this article is organized as follows. Section 2 develops the variational formulation and associated space-time Galerkin finite element approximation of the optimality conditions for the transport source inversion problem. Section 3 presents results of numerical experiments that study the performance of the inverse method, while in Section 4 we overview the high-level finite element toolkit Sundance, which was used for the numerical implementation. Finally, we offer conclusions in Section 5.

2 Variational formulation and finite element approximation

In this section, we derive the first-order optimality conditions for the source inversion problem. First we state a weak form of the transport equation (to which we refer as the *forward*, or *state* problem). Then we formulate the inverse problem of estimating the contaminant source term of the forward problem, given measurements of the contaminant concentration at a finite number of distinct points. Next, we derive the system of first-order optimality conditions in weak and strong forms.

These are necessary conditions for the solution to the inverse problem. We conclude the section with a discussion of a space-time finite element approximation of these optimality conditions, and comment on algebraic solver issues.

2.1 The forward problem

Given a known velocity field $\mathbf{v}(\mathbf{x}, t)$, the transport equation for the concentration $c(\mathbf{x}, t)$ in a domain Ω is given by the standard convection-diffusion equation

$$\begin{aligned} \frac{\partial c}{\partial t} - k\Delta c + \nabla c \cdot \mathbf{v} + f &= 0 \quad \text{in } \Omega \times (0, T), \\ \frac{\partial c}{\partial n} &= g \quad \text{on } \Gamma_N \times (0, T), \quad c = 0 \quad \text{on } \Gamma_D \times (0, T), \\ c &= c_0 \quad \text{in } \Omega \quad \text{at } t = 0, \end{aligned} \quad (1)$$

where $k > 0$ is the molecular diffusion (assumed constant), $f(\mathbf{x}, t)$ the source term that drives the concentration in the system, $c_0(\mathbf{x})$ the initial concentration distribution, Γ_N the portion of the boundary on which Neumann conditions $g(\mathbf{x}, t)$ are prescribed, and Γ_D the portion of the boundary with homogeneous Dirichlet conditions. For notational convenience we introduce the function space $V := \{c \in H^1(\Omega) : c = 0 \text{ on } \Gamma_D\}$ and the bilinear forms

$$\begin{aligned} a(f_1, f_2) &:= \int_{\Omega} \nabla f_1 \cdot \nabla f_2 \, d\Omega, & b(f_1, f_2) &:= \int_{\Omega} \nabla f_1 \cdot \mathbf{v} \, f_2 \, d\Omega, \\ s(f_1, f_2) &:= \int_{\Omega} f_1 \, f_2 \, d\Omega, & s(f_1, f_2)_{\Gamma} &:= \int_{\Gamma} f_1 \, f_2 \, d\Gamma. \end{aligned}$$

For this setup and for smooth boundaries, c will be in V , when f is in $H^{-1}(\Omega)$. A weak form of (1) is the following: find c in V such that $c = c_0$ at $t = 0$ and

$$\begin{aligned} s\left(\frac{\partial c}{\partial t}, \psi\right) + k a(\psi, c) + b(c, \psi) - k s(g, \psi)_{\Gamma_N} + s(f, \psi) &= 0, \\ \forall \psi \in V, \forall t \in (0, T). \end{aligned} \quad (2)$$

2.2 The inverse problem

We are interested in solving the following problem: given observations of the concentration $\{c_j^*\}_{j=1}^{N_r}$ at N_r locations $\{\mathbf{x}_j\}_{j=1}^{N_r}$ inside Ω , we wish to estimate the source term f that most closely reproduces the observed concentrations. The inverse problem is formulated as a constrained, regularized least squares parameter estimation problem:

$$\min_{f, c} \mathcal{J}(c, f) := \frac{1}{2} \sum_{j=1}^{N_r} \int_0^T s((c - c_j^*)^2, \delta(\mathbf{x} - \mathbf{x}_j)) \, dt + \frac{1}{2} \beta \mathcal{R}(f), \quad (3)$$

subject to the constraints given by the (weak form of the) convection-diffusion equation (2).

At the outset, we do not assume any *a priori* information about the structure of f in space or time. Upon discretization we will have as many unknowns as the dimension of the approximation space; for a space-time Galerkin approximation, for example this will equal the number of spatial grid points multiplied by the number of time steps. If the misfit between predicted and observed concentrations is measured throughout the domain, the problem can be viewed as a matching control problem, which is known to have a unique solution (by linearity of constraints and by strict convexity of the observation functional); see [4, 5] for theory and numerical examples for matching flow control problems. However, in most inverse problems, the observation data is limited. Typically, we have just a finite number (N_r) of observation locations at our disposal, and therefore, by a simple counting argument applied to the discretized problem, we see that the inverse problem is underspecified, i.e. limited measurements imply multiple solutions, and hence ill-posedness in the sense of Hadamard. Therefore some type of *regularization* is needed in order to obtain a well-posed problem. For this reason we have included $\mathcal{R}(f)$ in (3) to represent the regularization term. Some possible choices are

$$\mathcal{R}_2(f) := \int_0^T s(f, f) dt, \quad (4)$$

$$\mathcal{R}_e(f) := \int_0^T a(f, f) dt + \int_0^T s \left(\frac{\partial f}{\partial t}, \frac{\partial f}{\partial t} \right) dt, \quad (5)$$

$$\mathcal{R}_t(f) := \int_0^T s(\nabla f, \nabla f)^{\frac{1}{2}} dt + \int_0^T s \left(\frac{\partial f}{\partial t}, \frac{\partial f}{\partial t} \right)^{\frac{1}{2}} dt. \quad (6)$$

The first two regularizations are of Tikhonov type: \mathcal{R}_2 penalizes the space-time L^2 norm, and hence magnitude, of f , while \mathcal{R}_e penalizes the H^1 seminorm, and hence rough components, of f . Since the source f can be discontinuous, and since \mathcal{R}_e will smooth these discontinuities, we have also investigated the functional \mathcal{R}_t , which restricts f to the space of functions with bounded variation in space and time. Such *total variation* regularization penalizes oscillatory components while admitting discontinuities in f with penalty proportional to the jump (as opposed to an “infinite penalty,” as with \mathcal{R}_e). The tradeoff is that \mathcal{R}_t is nonquadratic and highly nonlinear, creating significant computational difficulties.

The amount of regularization is controlled by the parameter β , and its choice is related to the scaling of the spectrum of the observation (the sum of delta functions) and regularization operators. Choosing the proper value of β is a nontrivial task; there are a number of different approaches, and rather than review them we refer the reader to [13, Ch. 7].

2.3 Derivation of first-order optimality conditions

We reformulate the original constrained optimization problem as a system of (possibly nonlinear) partial differential equations by introducing an additional function, the Lagrange multiplier $\lambda \in V$, and a Lagrangian functional

$$\mathcal{L}(c, \lambda, f) := \mathcal{J}(c, f) + \int_0^T s\left(\frac{\partial c}{\partial t}, \lambda\right) + k a(\lambda, c) + b(c, \lambda) - k s(g, \lambda)_{\Gamma_N} + s(f, \lambda) dt. \quad (7)$$

Under mild conditions, solutions of (3) correspond to stationary points of (7). By taking variations with respect to c , λ , and f , we can derive a system of partial differential equations that characterize the first order optimality conditions for the optimization problem (3). These are also known as the Karush-Kuhn-Tucker (KKT) optimality conditions. Below we give details on the derivation.

2.3.1 State equation

Taking variations of the Lagrangian (7) with respect to the Lagrange multipliers and equating the result to zero for all admissible variations, i.e.

$$\partial_\lambda \mathcal{L}(c, \lambda, f) := \mathcal{L}(c, \lambda + \epsilon \hat{\lambda}, f)|_{\epsilon=0} = 0, \quad \forall \hat{\lambda} \in V,$$

we obtain the weak form of the state problem (2). Integration by parts then yields the strong form initial-boundary-value problem (1).

2.3.2 Adjoint equation

To derive the equation for the Lagrange multipliers, which is known as the *adjoint equation*, we take variations of the Lagrangian (7) with respect to c and equate the result to zero for all admissible variations. Namely, we take

$$\partial_c \mathcal{L}(c, \lambda, f) := \mathcal{L}(c + \epsilon \hat{c}, \lambda, f)|_{\epsilon=0} = 0, \quad \forall \hat{c} \in V,$$

which gives the weak form of the adjoint equation:

$$\sum_{j=1}^{N_r} \int_0^T s(c - c^*, \hat{c} \delta(\mathbf{x} - \mathbf{x}_j)) dt + \int_0^T s\left(\lambda, \frac{\partial \hat{c}}{\partial t}\right) + b(\hat{c}, \lambda) + k a(\lambda, \hat{c}) dt = 0, \quad \forall \hat{c} \in V. \quad (8)$$

By construction, the Lagrange multiplier satisfies homogeneous Dirichlet boundary conditions on Γ_D . To obtain the remaining conditions on λ , we integrate by parts;

in time for the term $\int_0^T s(\lambda, \partial \hat{c} / \partial t) dt$, and in space for the convective and diffusive terms. From the time integration we have

$$\int_0^T s(\lambda, \frac{\partial \hat{c}}{\partial t}) dt = \int_{\Omega} \lambda \hat{c}|_{t=0}^{t=T} d\Omega - \int_0^T s(\hat{c}, \frac{\partial \lambda}{\partial t}) dt.$$

Since $\hat{c} = 0$ at $t = 0$ but is arbitrary at $t = T$, we obtain the *final condition* $\lambda = 0$ at $t = T$. In space, using the vector identities

$$\nabla \cdot ((\hat{c} \mathbf{v}) \lambda) = \lambda \nabla \cdot (\hat{c} \mathbf{v}) + \nabla \lambda \cdot \mathbf{v} \hat{c} = \nabla c \cdot \mathbf{v} \lambda + \hat{c} (\nabla \cdot \mathbf{v}) \lambda,$$

and the divergence theorem, and assuming that the velocity field is incompressible, i.e. $\nabla \cdot \mathbf{v} = 0$, we arrive at

$$\int_0^T b(\hat{c}, \lambda) dt = \int_0^T -b(\lambda, \hat{c}) + s(\hat{c}, \lambda \mathbf{v} \cdot \mathbf{n})_{\Gamma} dt,$$

which combined with

$$\int_0^T k a(\lambda, \hat{c}) dt = \int_0^T \int_{\Omega} -\hat{c} \nabla \cdot k \nabla \lambda d\Omega + s(\hat{c}, k \frac{\partial \lambda}{\partial n}) dt,$$

results in the mixed condition $(k \nabla \lambda + \lambda \mathbf{v}) \cdot \mathbf{n} = 0$ on Γ_N . Therefore, the strong form for the adjoint equation is given by

$$\begin{aligned} -\frac{\partial \lambda}{\partial t} - k \Delta \lambda - \nabla \lambda \cdot \mathbf{v} &= \sum_{j=1}^{N_r} (c_j^* - c_j), \quad \text{in } \Omega \times (0, T) \\ (k \nabla \lambda + \lambda \mathbf{v}) \cdot \mathbf{n} &= 0, \quad \text{on } \Gamma_N \times (0, T), \quad \lambda = 0 \quad \text{on } \Gamma_D \times (0, T), \\ &\lambda = 0 \quad \text{in } \Omega \quad \text{at } t = T. \end{aligned} \quad (9)$$

Notice that this is a final value problem (it has a final, rather than initial, condition), and is linear in the unknowns c and λ .

2.3.3 Source equation

Taking variations of the Lagrangian (7) with respect to f , we obtain the *source equation*. Here we present a regularization formulation that has the form of the total variation functional \mathcal{R}_t in space, but the H^1 Tikhonov \mathcal{R}_e functional in time. Let

$$w(f) := \frac{1}{2(\nabla f \cdot \nabla f)^{\frac{1}{2}}},$$

then,

$$\partial_f \mathcal{R}(f, \hat{f}) := \int_0^T \int_{\Omega} w(f) \nabla f \cdot \nabla \hat{f} d\Omega dt + \int_0^T s(\frac{\partial f}{\partial t}, \frac{\partial \hat{f}}{\partial t}) dt. \quad (10)$$

Taking variations of the Lagrangian with respect to f we obtain

$$\partial_f \overline{\mathcal{L}} = \beta \partial_f \mathcal{R}(f, \hat{f}) - \int_0^T s(\lambda, \hat{f}) dt = 0, \quad \forall \hat{f} \text{ in } V. \quad (11)$$

By carrying out integration by parts in space and time we can derive the boundary conditions and the strong form of the source equation:

$$\begin{aligned} -\beta (\nabla \cdot w(f) \nabla f + \frac{\partial^2 f}{\partial t^2}) - \lambda &= 0, \quad \text{in } \Omega \times (0, T) \\ \frac{\partial f}{\partial n} &= 0 \quad \text{on } \Gamma \times (0, T), \quad \frac{\partial f}{\partial t} = 0, \quad \text{at } t = 0, t = T. \end{aligned} \quad (12)$$

Notice that while the forward and adjoint problems are evolution equations (in fact mixed elliptic-hyperbolic), the source problem (12) is an equilibrium problem, in fact an elliptic problem, over the space-time domain $\Omega \times (0, T)$. The differential operator is a $(d + 1)$ -dimensional nonlinear Poisson operator in space-time, and the problem has Neumann boundary conditions on all boundaries, including those associated with the time direction. Had we chosen the pure H^1 Tikhonov regularization \mathcal{R}_e defined in (5), the elliptic operator governing the the source problem would reduce to the $(d + 1)$ -dimensional Laplacian, and (12) would become a linear PDE problem. Finally, the choice of the L^2 regularizer \mathcal{R}_0 would result in an algebraic source equation.

One final remark: to guarantee solvability of (12), we have to modify \mathcal{R} , since $w(f)$ is unbounded where the gradient of f is zero. Therefore, we modify the regularization functional by introducing an additional (small) parameter α so that

$$\mathcal{R}^\alpha(f; \alpha) = \int_0^T \int_\Omega (\nabla f \cdot \nabla f + \alpha)^{\frac{1}{2}} d\Omega dt + \int_0^T s\left(\frac{\partial f}{\partial t}, \frac{\partial f}{\partial t}\right) dt. \quad (13)$$

The perturbed regularizer \mathcal{R}^α is now differentiable where $|\nabla f|$ is zero. The expression for the variation of \mathcal{R} with respect to f , (10), is still valid, but with $w(f)$ replaced by

$$w^\alpha(f; \alpha) := \frac{1}{2(\nabla f \cdot \nabla f + \alpha)^{\frac{1}{2}}}. \quad (14)$$

The choice of β and α has great effect on the accuracy of the solution and the conditioning of the problem. For a discussion see [1].

2.3.4 KKT optimality conditions

In the derivation of the KKT optimality conditions, we assumed the weak forms to hold a.e. in time, in accordance with a conventional finite-difference type approximation in time. However, the system of forward, adjoint, and source equations is elliptically coupled in space-time. Therefore, a natural choice for numerical solution is to discretize by space-time finite elements, based on a space-time weak

The fact that the optimality system (16) is approximated by a Galerkin finite element method does not imply that the state or adjoint convection diffusion problems must be *individually* approximated by Galerkin methods. It means only that the test space for the state equation must be taken as the trial space for the Lagrange multiplier, and the test space for the adjoint equation will be the trial space for the concentration. This allows use of an appropriate Petrov-Galerkin approximation for the state equation; for example, we could have employed a streamline diffusion method, as may be preferred for a convectively-dominated transport problem, and still maintained consistency with the approximation in (16).

Several of the numerical results presented in Section 3 are for a steady state approximation to the time-dependent optimality system above. In this case, the weak form of the KKT optimality system (assuming zero Neumann boundary conditions on the state problem, i.e. $g = 0$) is given by:

Find $f, c, \lambda, \in V$ such that

$$\begin{aligned} k a(\hat{\lambda}, c) + b(c, \hat{\lambda}) + s(f, \hat{\lambda}) &= 0, \quad \forall \hat{\lambda} \in V \\ \sum_{j=1}^{N_r} (\delta(\mathbf{x} - \mathbf{x}_j))(c - c^*)\hat{c} + k a(\hat{c}, \lambda) - b(\lambda, \hat{c}) &= 0, \quad \forall \hat{c} \in V \quad (17) \\ \beta s(w^\alpha(f) \nabla f, \nabla \hat{f}) - s(\lambda, \hat{f}) &= 0, \quad \forall \hat{f} \in V. \end{aligned}$$

Finite element approximation of this system follows straightforwardly:

Find $f_h, c_h, \lambda_h, \in V_h$ such that

$$\begin{aligned} k a(\hat{\lambda}_h, c_h) + b(c_h, \hat{\lambda}_h) + s(f_h, \hat{\lambda}_h) &= 0, \quad \forall \hat{\lambda}_h \in V_h \\ \sum_{j=1}^{N_r} (\delta(\mathbf{x} - \mathbf{x}_j))(c_h - c^*)\hat{c}_h + k a(\hat{c}_h, \lambda_h) - b(\lambda_h, \hat{c}_h) &= 0, \quad \forall \hat{c}_h \in V_h \quad (18) \\ \beta s(w^\alpha(f_h) \nabla f_h, \nabla \hat{f}_h) - s(\lambda_h, \hat{f}_h) &= 0, \quad \forall \hat{f}_h \in V_h. \end{aligned}$$

Finally, we note that several of the numerical examples in Section 3 use the H^1 Tikhonov regularization (5), rather than the mixed total variation/Tikhonov regularizer defined by (13). In this case, the time-dependent and steady optimality systems (16) and (18) are still valid, provided we take $w^\alpha := 1$.

2.3.5 Solution Algorithm

The discrete optimality systems (16) and (18) constitute formidable systems to solve, particularly in the time-dependent case. A full discussion of solver issues is beyond the scope of this article, and will be presented in a follow-on article. However, several pertinent comments are in order. One way to avoid the large size and coupled nature of (16) and (18) is to effect a block-elimination of state and Lagrange multiplier variables, and reduce the system to one in just the source unknowns. We illustrate this idea in the context of the steady-state optimality system

(18), but the ideas are similar in the time-dependent case. Given an estimate of the source f_h , first solve for the concentration c_h from the state equation,

$$k a(\hat{\lambda}_h, c_h) + b(c_h, \hat{\lambda}_h) + s(f_h, \hat{\lambda}_h) = 0, \quad \forall \hat{\lambda}_h \in V_h;$$

then with c_h known solve the adjoint equation

$$\sum_{j=1}^{N_r} (\delta(\mathbf{x} - \mathbf{x}_j)(c_h - c^*)\hat{c}_h) + k a(\hat{c}_h, \lambda)_h - b(\lambda_h, \hat{c}_h) = 0, \quad \forall \hat{c}_h \in V_h$$

for λ_h ; and finally solve the “reduced” source equation

$$\beta s(w^\alpha(f_h) \nabla f_h, \nabla \hat{f}_h) - s(\lambda_h, \hat{f}_h) = 0, \quad \forall \hat{f}_h \in V_h.$$

to update f_h . This iteration can be accelerated by the conjugate gradient method (the reduced source equation is positive definite). For Tikhonov regularization, the reduced source system is linear, and only linear iterations are required. For total variation regularization, the reduced source system is nonlinear, and the inner linear iterations must be embedded within a nonlinear iteration, for example Newton’s method. This block-elimination is a special case of the reduced space Sequential Quadratic Programming method [11]. Since *each* linear iteration of this reduced space method requires the solution of state and adjoint transport PDEs, the method can be very expensive. On the other hand, the advantage is that structure is imparted onto the KKT optimality system (by decomposing into state, adjoint, and source solves); moreover, one can capitalize on available algorithms and software for the state and adjoint transport equation solves, and (nonlinear) Poisson solvers for the source equation.

Despite the advantages of this block-elimination, the requirement of solving state and adjoint systems at each iteration is often onerous, and we prefer to solve the entire KKT optimality system (16) or (18) simultaneously. Since these constitute very large systems, particularly for 3D or time-dependent problems, iterative solvers and good preconditioners are essential. We will not discuss these issues in detail, and instead direct the reader to the detailed descriptions of the inner linear iteration in [2] and the outer nonlinear iteration in [3]. The basic idea is that one can solve in the full space iteratively, but precondition via an approximate version of the reduced space method, in which state and adjoint operators are replaced by their own preconditioners, and the source operator can be approximated by ideas from quasi-Newton optimization theory. At least in the steady-state case, extremely efficient algorithms result, and one can often obtain the solution to the inverse problem in a small multiple (as little as 4 or 5) of the cost of solving the state equation. This was demonstrated in [2, 3] in the context of boundary optimal control of steady Navier-Stokes flows. Related algorithms are also discussed in [7].

Finally, regardless of whether we solve the full KKT system simultaneously or within the reduced space of source unknowns, in the 3D time-dependent case we

are faced with source problems that involve 4D (space plus time) Laplacian operators for H^1 Tikhonov regularization, and nonlinear 4D Poisson operators for total variation regularization. These are certainly challenging problems and require specialized solvers, especially on unstructured meshes.

3 Numerical results

In this section we provide sample numerical results to examine the performance of the inverse method on synthetic observational data. We provide results for solution of both the steady state (18) and time-dependent (16) discrete optimality systems, using both H^1 Tikhonov ($w^\alpha = 1$) and total variation (w^α given by (14)) regularization. Sensors that measure the concentration field are placed throughout the spatial domain on a regular array. We synthesize sensor readings by proposing a source f^* , and then solve the inverse problem to determine how well the source is reconstructed.

In all of the numerical experiments, the discretization in (18) and (16) is effected via piecewise linear finite elements in both space and (when applicable) time. We discretize the state, adjoint and source variables on the same regular grid consisting of linear triangles. Newton's method is used to solve the nonlinear systems that arise when total variation regularization is used; a simple backtracking line search is employed to ensure convergence. The linear systems that are produced either at each iteration of Newton's method, or once and for all when Tikhonov regularization is used, are solved iteratively using the biconjugate gradient stabilized method (BICGSTAB). A level-k incomplete LU factorization (ILUK) is used to construct a preconditioner. We invoke a convergence tolerance of 10^{-10} to terminate both the inner (linear) as well as outer (nonlinear) iterations.

Since we are using a variationally-based Galerkin approximation, it should be possible to obtain a symmetric linear(ized) KKT system. In fact, if the equation blocks are ordered as (adjoint, source, state), and the variable blocks ordered as (state, source, Lagrange multiplier), we do obtain a symmetric discrete system, as can be verified from (18) or (16). However, in this case, the (3,3) block, corresponding to the coefficient of the Lagrange multiplier in the state equation, is identically zero, and standard ILU preconditioning will fail. Therefore, we reorder the equations as (state, adjoint, source) and the unknowns as (state, Lagrange multiplier, source), which is the order implied by the reduced space method. This places the state, adjoint, and source operators on the main diagonal blocks, and prevents failure (and indeed leads to reasonable performance) of the ILUK preconditioner. However, this ordering creates a nonsymmetric linear system, which is why we use BICGSTAB.

In all of the examples below, we choose the regularization constant $\beta = 10^{-5}$, and (where appropriate) the total variation parameter $\alpha = 0.1$. The spatial domain is a

square of dimension $2L \times 2L$, with $L = 8$, and for time dependent problems, T is taken as 10. The diffusion constant k is taken as 1. We choose a velocity field \mathbf{v} given by a Poiseuille flow, i.e. $v_x(y) = (L + y)(L - y)/(L^2)$, $v_y = 0$. This has maximum value of 1, at the centerline, and thus the Peclet number, $Pe = 2Lv_0/k$, based on this value is 8. To create a sequence of problems with increasing Pe , we simply scale up the velocity field.

In the subsections below, we study the dependence of the quality of inversion on the number of sensors, the type of regularization, the ratio of convection to diffusion, and the time dependence of the source. We also examine the convergence rate of the finite element approximation of the inverse problem to the exact solution.

3.1 Effect of number of sensors

In this subsection we study the influence of the number of sensors on the quality of the reconstructed source. The steady-state problem (18) is solved. Boundary conditions on the concentration c are homogeneous Dirichlet on the left boundary and homogeneous Neumann conditions on the right, top, and bottom sides. The target source function $f^*(x, y)$ consists of two concentrated Gaussians with different peak magnitudes and location. We discretize the problem on a 40×40 grid (which means we have 3042 linear triangular elements, 1600 grid points, and a total of 4680 concentration, multiplier, and source unknowns). We use Tikhonov H^1 regularization, and solve the same problem with three sensor arrays of increasingly finer density. The array consists of uniformly-spaced sensors that cover the domain. Figure 1 plots the reconstructed source for arrays of 4×4 , 10×10 , and 40×40 sensors. The last case, shown at the bottom of the figure, is indistinguishable from the target source, which is why we do not display f^* . Inversion of observations from the 4×4 sensor array is unable to resolve the location or magnitude of the second source, while the first source is greatly smoothed. The 10×10 sensor array leads to a much better inversion; the locations and general shape are well-resolved. On the other hand, the inversion is less successful in establishing the magnitude of the two Gaussians. Finally, inversion with the 40×40 sensor array leads to a reconstruction that is nearly indistinguishable from the exact source. Since sensors are placed at every grid point, the inverse problem is well-posed in the absence of the regularization term, and we could have obtained the exact source by setting the regularization parameter to zero. Since here we are using $\beta = 10^{-5}$, the inversion operator is dominated by the (full-rank) least-squares term, and the influence of the regularization matrix (a discrete Laplacian stiffness matrix) is minimal, leading to source reconstruction that is almost exact.

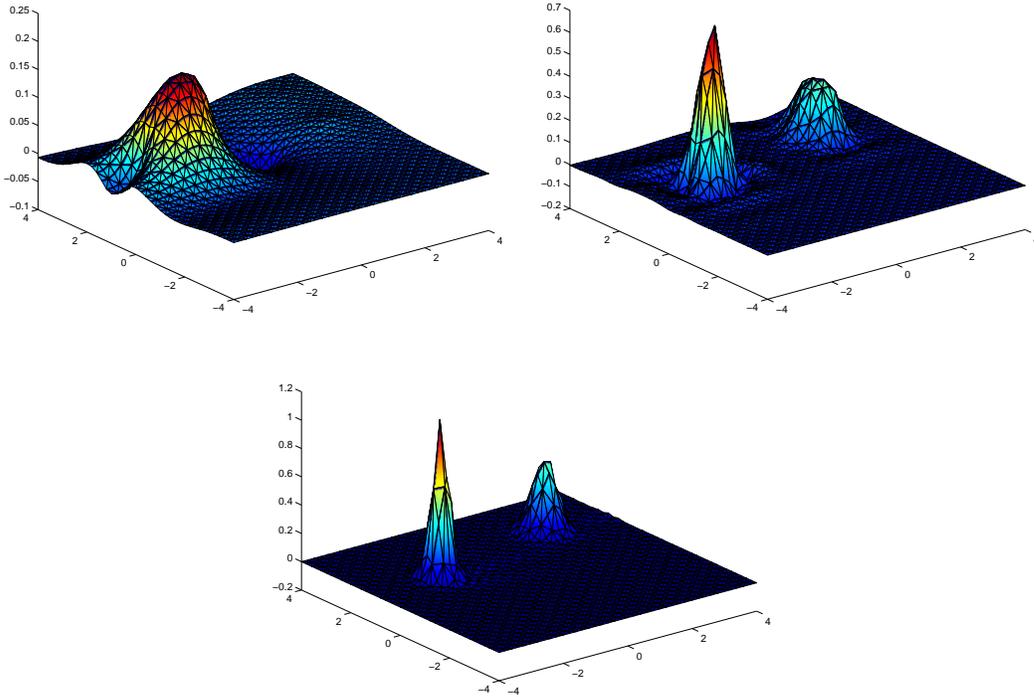


Fig. 1. Effect of sensor density on source inversion. Steady transport, Tikhonov H^1 regularization, 40×40 grid, flow from upper left to lower right. Top left, 4×4 sensor array; top right, 10×10 sensor array; bottom, 40×40 sensor array.

3.2 Effect of regularization type

The numerical experiments of the previous subsection used a smooth source (a sum of Gaussians), for which Tikhonov regularization performs reliably. In our next set of experiments, we use a nonsmooth target source and compare the effectiveness of Tikhonov and total variation regularization. Boundary conditions and grid size are as in Subsection 3.1. The sensors are located on a 10×10 array. The target source function consists of two “point” sources, that actually have constant strength over a patch of 4 cells (i.e. 8 triangles), and taper linearly to zero in neighboring elements; this is depicted in the upper left image of Figure 2. The results of inversion using H^1 Tikhonov regularization are shown in the upper right figure. As is clear from the figure, Tikhonov does a poor job in reconstructing the source. Both shapes are severely smoothed, and one can observe substantial oscillations near the interface of the right source. In contrast, the Tikhonov regularization shown in Figure 1 on the same sensor array is much more effective at identifying a similarly concentrated, but smooth source (a Gaussian). On the other hand, total variation regularization, as shown in the bottom image of the figure, results in a much better representation of the target source. While the magnitude is still incorrect, the shapes are well-localized, with significantly less smoothing. We remark that total variation regularization is effective at reconstructing discontinuous functions (i.e. in L^2), but

in this case we are faced with target sources that approach a point source (i.e. in H^{-1}), for which total variation produces an unbounded penalty. We should also keep in mind that total variation regularization results in a nonlinear system of optimality conditions, which are significantly more expensive to solve than for the Tikhonov case.

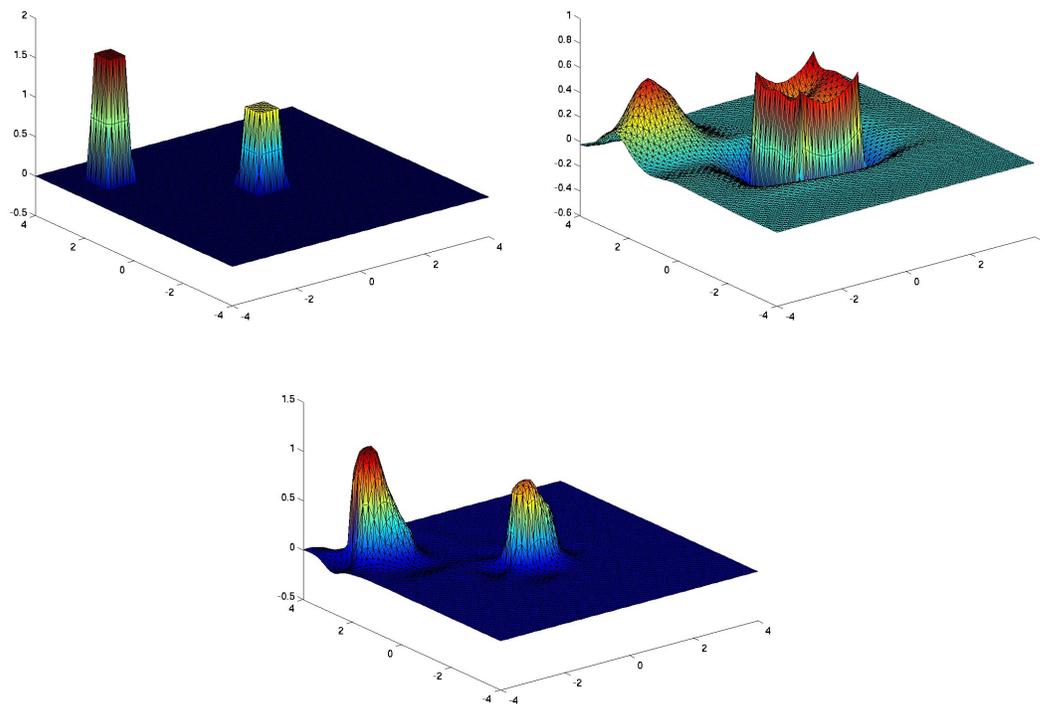


Fig. 2. Effect of regularization type on source inversion. Steady transport, 40×40 grid, 10×10 sensor array, flow from upper left to lower right. Upper left image, target source function; upper right image, inversion with Tikhonov regularization; bottom image, inversion with total variation regularization.

3.3 Effect of Peclet number

In this subsection, we examine the effect of increasing Peclet number on the quality of the inverted source. In particular, we study the ability to invert for both diffuse and concentrated sources as the transport becomes increasingly convection-dominated. For these experiments, we solve the steady form of the KKT optimality conditions (18) and impose homogeneous Dirichlet boundary conditions on the concentration c on all boundaries. Figure 3 shows results obtained when the target source is taken as a smooth, diffuse function (a Gaussian). The flow is from left to right. We use a 160×160 grid, and the sensors are located on a 6×6 uniformly-spaced array. The bottom image in the figure depicts the color contours of the target source function. The top three rows show color contours of the concentration field

(left column) and source reconstruction (right column) for three different Peclet numbers, namely 0, 8, and 80. The left column of images illustrates the increasingly convection-dominated concentration field. The right column demonstrates that for this problem the method is able to reconstruct a source that is essentially identical to the target source, independent of the Peclet number. The 6×6 sensor array apparently places a sufficient number of sensors in the concentration field wake to identify accurately the source field.

A different picture emerges when a highly concentrated Gaussian is chosen as the target. Figure 4 presents the results. The bottom image displays contours of the target source. We compare inversion for Pe 0 (pure diffusion, discretized on a 160×160 grid) and Pe 800 (strong convection, discretized on a 320×320 grid). The left column refers to the pure diffusion case, and the right column to the large convection case, as can be inferred from the concentration contours shown in the first row. The second and third rows show the results of inversion using 6×6 and 21×21 sensors arrays, respectively. The 6×6 sensor array is unable to resolve the source, neither for Pe 0 nor 800, as shown in the second row of the figure. However, a 21×21 sensor array is able to accurately locate the source for pure diffusion; the reconstructed source is indistinguishable from the target. The opposite conclusion is reached with the Pe 800. In this case, apparently even a 21×21 sensor array does not place sufficient sensors in the concentration wake to produce an accurate inversion; the image shows that the recovered source is a highly-smoothed approximation of the target.

3.4 Time dependent inversion

In this subsection we demonstrate the ability of the inverse formulation to reconstruct a time-dependent source. Recall that we must solve the space-time-coupled optimality conditions (16) over the space-time box $0 \leq x \leq L, 0 \leq y \leq L, 0 \leq t \leq T$. Homogeneous Neumann boundary conditions and zero initial conditions are imposed on c . H^1 Tikhonov regularization, in space and time, is used. The three-dimensional space-time volume is discretized on a $30 \times 30 \times 30$ grid. Sensors are placed on a $10 \times 10 \times 10$ uniform array (i.e. the sensors locations do not change over time). The target is a concentrated Gaussian that moves in time across the spatial domain from the upper left to the lower right, as shown in Figure 5. The right column depicts the target source; the four rows correspond to snapshots of the source at $t = 0, 3.3, 6.7, 10$, respectively. The left column of the figure displays four snapshots (at the same time instants) of the reconstructed source. By comparing left and right columns, we see that the reconstruction is excellent inside the interval $0 < t < 10$, and deteriorates somewhat near the endpoints of the interval, i.e. $t = 0$ and $t = 10$. Actually, considering the rather coarse grid that is used (30 grid points or time steps in each direction), even the reconstruction near the initial and final conditions can be considered quite good. It is tempting to expect that the

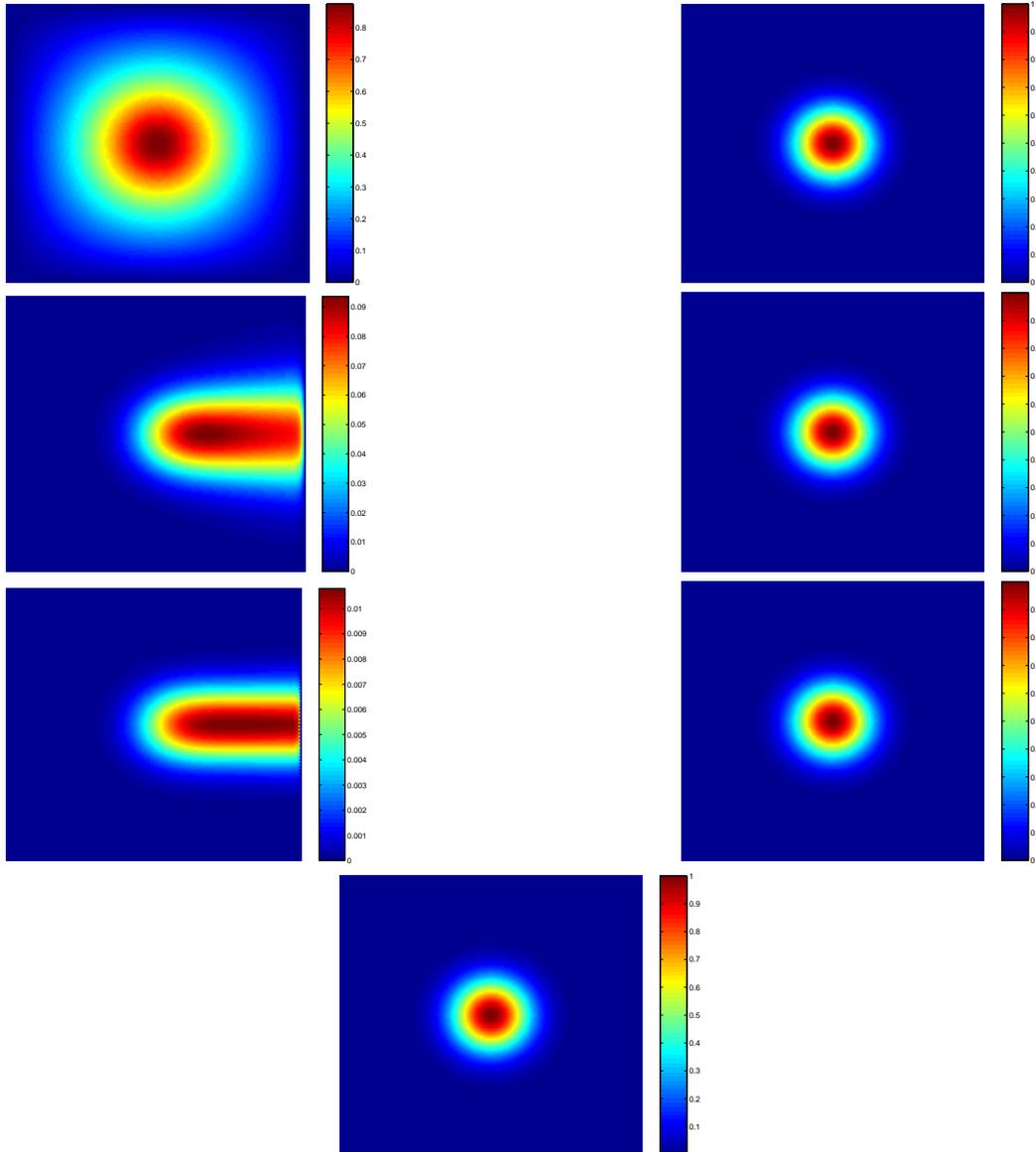


Fig. 3. Effect of Peclet on the quality of the inverted source for a smooth, diffuse target source. Steady problem, Tikhonov H^1 regularization, homogeneous Dirichlet boundary conditions on c , 6×6 sensor array, 160×160 grid. Bottom image shows target source. First row corresponds to Pe 0; second, to Pe 10; third, to Pe 100. Left column show contours of concentration field; right column shows reconstructed source.

quality of the inversion will worsen steadily over time, since the forward problem evolves from initial conditions. Recall, however, that the source equation (whose linearized operator determines the character of the inverted source) is a boundary value problem in space-time; there is no reason, therefore, to expect evolutionary behavior for the inverted source. On the contrary, when the source is near spatial boundaries (i.e. at $t = 0$ and $t = 10$), we cannot expect that the reconstruction will be as accurate as when it is in the interior, since there are fewer sensors in or near its support.

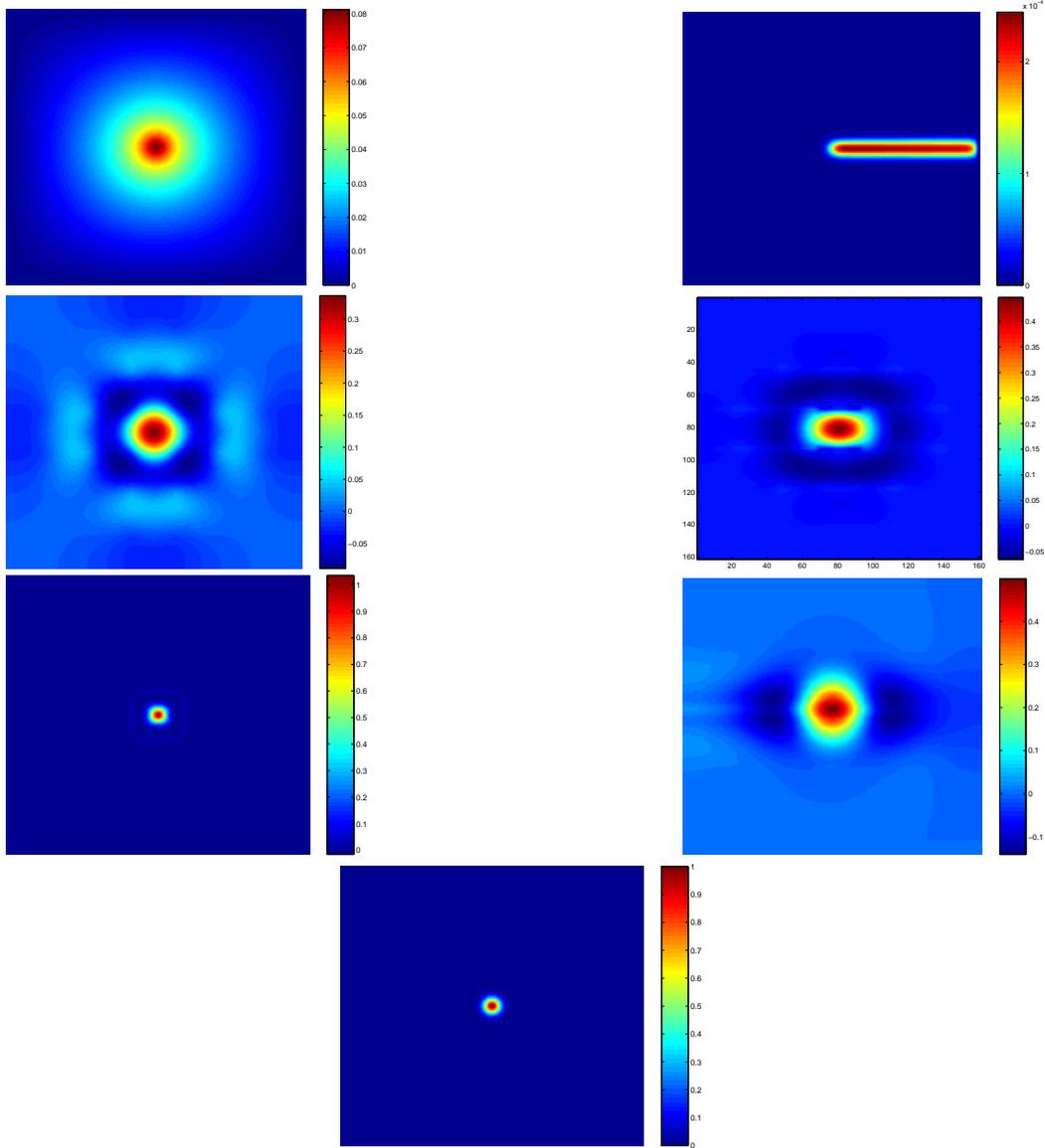


Fig. 4. Effect of Peclet number on the quality of the inverted source, for a concentrated source. Steady state transport, Tikhonov H^1 regularization, homogeneous Dirichlet boundary conditions on c . Target source is concentrated Gaussian, shown in bottom image. Left column corresponds to Pe 0 (discretized on 160×160 grid), right to Pe 800 (discretized on 320×320 grid). Top row shows concentration contours; middle shows reconstructed sources with 6×6 sensor array; third row shows reconstructed sources with 21×21 sensor array.

3.5 Convergence Rates

In this subsection we provide a study of the accuracy of the finite element approximation of the optimality system (18). We measure the rate of convergence of the finite element approximation of the reconstructed source, f_h , to the source that solves the inverse problem (17) exactly. Unfortunately, this is not f^* , the source

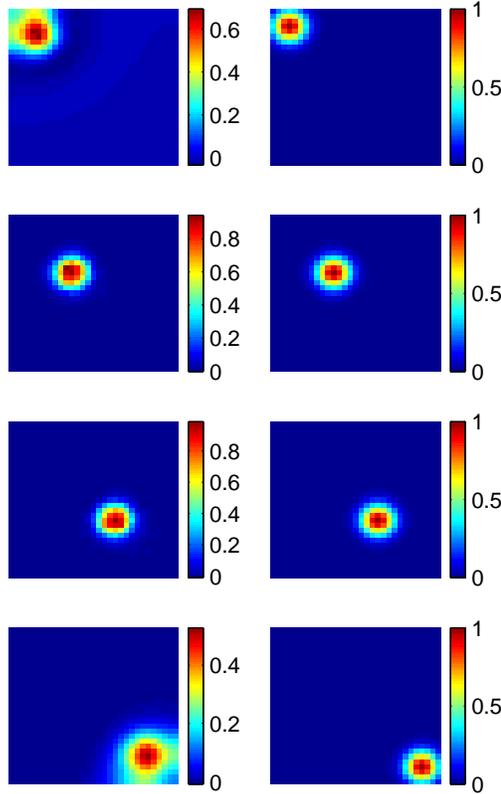


Fig. 5. Time dependent source inversion example, H^1 Tikhonov regularization, concentrated moving Gaussian source, $30 \times 30 \times 30$ space-time grid, $10 \times 10 \times 10$ sensor array. Right column shows color contours of the target source; left column shows reconstructed source. Rows correspond to four snapshots in time ($t = 0, 3.3, 6.7, 10$).

that was used to synthesize the sensor observations $c^*(\mathbf{x}_j)$; the inclusion of H^1 Tikhonov regularization (i.e. the term involving β in (17)) has modified the exact solution of the optimality system. The reason is as follows: the null space of the second variation of the least squares term (i.e. the first term in (3)) contains oscillatory source functions (a coarse spacing of sensors provides sufficient data to recover just the smooth components of the source). The inverse operator is thus rank deficient, which is why we add the regularization term (the term involving β) in (3). For simplicity, let us assume that \mathcal{R} is chosen to be the H^1 Tikhonov regularizer. The second variation of this term is just the Laplacian; therefore, we can interpret the regularization term as one that penalizes the most oscillatory components of f most strongly (since the largest eigenvalues of the Laplacian are associated with the most oscillatory eigenfunctions). The oscillatory functions lying in the null space of the least squares term will now have positive eigenvalues associated with them, the inverse operator (i.e. the second variation of \mathcal{J} with respect to the reduced space of source functions f) becomes positive definite, and the inverse problem becomes well posed. Unfortunately, the Laplacian regularizer also penalizes smooth eigenfunctions of f , which lie in the range space of the least squares term, albeit the penalty is much smaller than for the oscillatory components. We have some de-

grid	$\ f_h - f^*\ _2$
16×16	1.265×10^{-2}
32×32	2.163×10^{-3}
64×64	5.126×10^{-4}
128×128	1.264×10^{-4}
256×256	3.160×10^{-5}

Table 1

Convergence of finite element approximation of reconstructed source f^h to the exact reconstructed source f^* . Linear elements, Gaussian source. Convergence rate is optimal.

gree of control through the choice of β , which must be chosen large enough so that rough components of f (those that are poorly-determined by the observations) are filtered as much as possible, but small enough so that smooth components (those that are well determined by the observations) are as unaffected as possible.

To compute the convergence rate of the finite element approximation, however, we require an exact solution of the inverse problem. Therefore, we shall create a regularizer that renders the solution to the inverse problem unique and equal to f^* . This can be achieved by choosing the regularizer

$$\mathcal{R}(f) = \int_{\Omega} (\nabla f - \nabla f^*) \cdot (\nabla f - \nabla f^*) d\Omega.$$

We see that the choice $f = f^*$ minimizes both the regularization term and the least squares term (because $c = c^*$) in (3), and therefore f^* is the solution to this suitably-regularized inverse problem. Moreover, this choice of regularizer adds only the forcing term $\beta s(\nabla f_h^*, \nabla \hat{f}_h)$ to the source equation of the optimality system (17), and does not change the linear operator. This gives us a suitable test of finite element convergence rate.

Table 1 lists the error in the finite element approximation of f^* for a sequence of five increasingly finer meshes. The exact source f^* is taken as a Gaussian. We use the standard L^2 norm error

$$\|f_h - f^*\|_2^2 = \int_{\Omega} (f_h - f^*)^2 d\Omega.$$

The results exhibit a quadratic rate of convergence. Since we are using linear elements, this rate is optimal.

4 Sundance implementation

The numerical examples in this paper were implemented using a general-purpose, high-level, finite element toolkit called Sundance [8]. Sundance accepts a system

of coupled PDEs and boundary conditions written in symbolic form that is close to the notation in which a scientist or engineer would normally write them with pencil and paper. Each function appearing in this symbolic description is annotated with a specification of the finite-element basis with which that object will be discretized. This information, along with a mesh, is then used by Sundance to assemble the implied discrete operators and vectors. This matrix assembly step is fully and transparently parallelized. These capabilities make Sundance a powerful rapid prototyping and algorithmic research tool. With Sundance's flexible symbolic interface, it is straightforward to assemble and solve the discrete KKT systems (16) and (18). This section provides a brief introduction to Sundance.

4.1 Overview of Sundance

Sundance is a collection of C++ classes that can be used as building blocks to assemble a finite element computation. The PDEs and boundary conditions are written in terms of symbolic expression objects, or `Exprs`, the subtypes of which can represent mathematical concepts such as functions, test functions, and differential operators. The problem geometry is given in terms of a `Mesh` object which represents an unstructured mesh. Subdomains of a mesh are represented by `CellSet` objects, which are operators that can identify cells on which an equation or boundary condition is to be applied. Linear operators, vectors, and solvers are implemented using the TSF [9] linear algebra components. Details such as memory management and parallelism are handled transparently.

The guiding design principle behind Sundance is that the user should be able to specify a finite element problem in terms of a symbolic statement of a variational form and boundary conditions, augmented by high-level discretization specifiers such as basis function type and quadrature method. The user should be shielded from all details of low-level bookkeeping of elements, nodes, and degrees of freedom. A sophisticated user will want fine control over discretization, but that control should be done through high-level directives. For example, in Sundance the ordering of block rows is specified by the order of a list of test functions objects in a single constructor argument.

We must emphasize that for performance reasons, the high-level objects used for problem specification are not used for numerical calculations. Rather, they are used to marshal a set of internal objects that can be used for efficient calculations. Thus, Sundance can provide a high-level interface with a minimal performance loss. To illustrate basic Sundance objects and their use, below we show code fragments for a forward convection-diffusion problem on a 2D rectangle.

4.2 Defining the problem geometry

We begin by reading a mesh and defining boundary surfaces. Sundance has no built-in general-purpose meshing capability; in order to work with 3rd-party meshing components it defines interfaces for mesh generators and mesh readers. Here we create a reader object capable of reading a file in Shewchuk's Triangle [12] format, and read a mesh.

```
MeshReader reader = new ShewchukMeshReader( ``square`` );
Mesh mesh = reader.getMesh();
```

Subdomains such as boundary surfaces or internal regions are identified with `CellSet` objects, which can be created in a number of ways. Here, we identify subsets of the boundary that satisfy conditions on the coordinates. The coordinates are expressions, or `Expr` objects, and may be used in any symbolic expression.

```
/* define coordinate functions for x and y coordinates */
Expr x = new CoordExpr(0);
Expr y = new CoordExpr(1);

/* define cells sets for each of the four sides of the rectangle */
CellSet boundary = new BoundaryCellSet();
CellSet left = boundary.subset( x == -L );
CellSet right = boundary.subset( x == L );
CellSet bottom = boundary.subset( y == -L );
CellSet top = boundary.subset( y == L );
```

The `CellSet` class supports logical operations such as union, intersection, and exclusion. In many problems, numerical or character labels will be associated with nodes or edges, so `CellSet` has the ability to identify labeled regions.

4.3 Setting up the symbolic problem

The test and unknown functions and differential operators are represented as `Expr` objects. The test and unknown functions are constructed with arguments giving the type and order of their bases.

```
/* create symbolic objects for test and unknown functions */
Expr v = new TestFunction(new Lagrange(2));
Expr u = new UnknownFunction(new Lagrange(2));

/* create symbolic differential operators */
Expr dx = new Derivative(0);
Expr dy = new Derivative(1);
```

```
Expr grad = List(dx, dy);
```

The velocity field can be written in terms of coordinate functions.

```
Expr vx = (L+y)*(L-y);  
Expr vy = 0.0;  
Expr velocity = List(vx, vy);
```

Finally, we can use these components to write the weak PDE and boundary conditions in symbolic form

```
/* Write symbolic weak equation */  
Expr eqn = Integral(-(grad*v)*(grad*u)/peclet  
  - v*(velocity*grad)*u + v*source);  
  
/* write dirichlet BCs */  
EssentialBC bc = EssentialBC(left, u*v);
```

Had we wanted to use a non-default quadrature scheme for any of these terms, we could have supplied a `QuadratureFamily` object as an additional argument to the `Integral` method or the `EssentialBC` constructor.

4.4 Solving the problem

We next build a `StaticLinearProblem` object which is responsible for building the operators implied by the symbolic description and mesh.

```
StaticLinearProblem prob(mesh, eqn, bc, v, u);
```

Finally, we can specify a solver and solve the problem

```
TSFPreconditionerFactory precondition = new ILUKPreconditionerFactory(1);  
TSFLinearSolver solver = new BICGSTABSolver(precondition, 1.e-14, 500);  
  
Expr soln = prob.solve(solver);
```

The solution is returned as a symbolic object; this allows the use of previous solutions in subsequent problems, as occurs in time-marching or nonlinear problems.

4.5 Implementation of the source inversion problems

The source inversion problems that are the subject of this paper are more complex than the simple forward problem shown above, so space does not permit a complete

exposition of their code. However, some comments on the Sundance implementation of the source inversion problems are in order:

- The sensor locations are specified in terms of `CellSet` objects, and then the integrations involving sensor measurements are done over these `CellSets`.
- With TV regularization, the problem is nonlinear and must be solved by iterating solutions of a linearized problem. Sundance provides high-level facilities for automatically linearizing a problem at the symbolic level, or alternatively the linearized problem can be written by hand using the Sundance symbolic components.
- As mentioned in Section 2, block row ordering is critical for the effectiveness of the ILU preconditioner used here. In a multivariable problem, a Sundance user specifies row and column ordering through the ordering of the lists of test and unknown functions given to the `StaticLinearProblem` constructor.

5 Conclusions and future work

In this article we have considered the inverse problem of determining an arbitrary source function in both steady-state and time-dependent convective-diffusive transport, given a velocity field and measurements of a concentration field at a finite number of points. To address ill-posedness of the problem, we employed both H^1 Tikhonov and total variation regularization. We presented a variational weak formulation of the first order optimality system, which includes the initial-boundary value state problem, the final-boundary value adjoint problem, and the space-time boundary value source problem. The weak form leads naturally to a Galerkin space-time finite element method that discretizes the space-time volume. A series of numerical experiments, implemented using the high-level finite element toolkit Sundance, leads us to the following conclusions:

- Sundance is sufficiently rich and flexible to permit rapid and easy implementation of finite element methods for many inverse problems involving PDEs.
- Increasing sensor coverage leads as expected to better identification of source strength and distribution. A uniformly distributed 10×10 sensor array provides quite reasonable reconstruction of the source field for rather concentrated Gaussian sources.
- H^1 Tikhonov regularization works well for smooth sources. For discontinuous sources, it performs poorly, smearing the source and leading to oscillations in the vicinity of the interface. Total variation regularization results in a much higher quality inversion, but leads to a highly nonlinear inverse problem.
- As the transport becomes increasingly convection-dominated, the ability to reconstruct a diffuse source is unhindered, for a sensor array of reasonable density (6×6).
- For a concentrated (nearly point) source, increasing the Peclet number results in

- a deterioration in the quality of the reconstructed source.
- The space-time variational formulation and associated finite element method is capable of tracking a moving concentrated source across a domain. However, introduction of the time dimension into the inverse problem results in a need to solve a $d + 1$ -dimensional elliptic problem over the space-time volume, where d is the space dimension. In three space dimensions, this poses a significant computational challenge.
- For reconstruction of a smooth source, we observe an optimal convergence rate of the finite element approximation to the exact solution (i.e. we obtain a rate of two for linear elements).

We are pursuing applications of the formulation presented in this article to problems of identification and remediation of hazardous contaminants. One setting of interest is the control of an HVAC system in a public facility, such as an airport, that has been attacked by releases of airborne chemical or biological agents. Solution of the source inversion problem, given observations from distributed sensors as well as modeled air flow fields, provides estimates of the source strength and distribution, as well as the distributed concentration field away from the observation points. These estimates are fed into an optimal control solver—with both contaminant transport and Navier-Stokes equations as constraints—that aims to control boundary velocities at vent locations to reduce the contaminant hazard as rapidly as possible. We have only briefly mentioned solver issues in this article. Clearly, scalable and robust solvers for the space-time KKT optimality equations are a critical component of an effective system that is able to respond to such threats in near real-time, and this constitutes a topic of focused interest for us.

References

- [1] V. Akcelik, G. Biros, and O. Ghattas. Parallel multiscale Gauss-Newton-Krylov methods for inverse wave propagation. In *Proceedings of IEEE/ACM SC2002 Conference*, Baltimore, MD, November 2002.
- [2] G. Biros and O. Ghattas. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part I: The Krylov-Schur solver. Technical report, Laboratory for Mechanics, Algorithms, and Computing, Carnegie Mellon University, 2000. Available from www.cs.cmu.edu/~oghattas.
- [3] G. Biros and O. Ghattas. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part II: The Lagrange-Newton solver, and its application to optimal control of steady viscous flows. Technical report, Laboratory for Mechanics, Algorithms, and Computing, Carnegie Mellon University, 2000. Available from www.cs.cmu.edu/~oghattas.
- [4] M.D. Gunzburger and S. Manservigi. The velocity tracking problem for

- Navier-Stokes flows with bounded distributed controls. *SIAM Journal on Control and Optimization*, 37(6):1913–1945, 1999.
- [5] M.D. Gunzburger and S. Manservigi. Analysis and approximations of the velocity tracking problem for Navier-Stokes flows with distributed control. *SIAM Journal on Numerical Analysis*, 37(5):1481–1512, 2000.
- [6] K.R. Gurney, R.M. Law, A.S. Denning, P.J. Rayner, D. Baker, P. Bousquet, L. Bruhwiler, Y.H. Chen, P. Ciais, S. Fan, I.Y. Fung, M. Gloor, M. Heimann, K. Higuchi, J. John, T. Maki, S. Maksyutov, K. Masarie, P. Peylin, M. Prather, B.C. Pak, J. Randerson and J. Sarmiento, S. Taguchi, T. Takahashi, and C.W. Yuen. Towards robust regional estimates of CO₂ sources and sinks using atmospheric transport models. *Nature*, 415(6872):626–630, 2002.
- [7] E. Haber and U. Ascher. Preconditioned all-at-once methods for large, sparse parameter estimation problems. *Inverse Problems*, 2001.
- [8] K. Long. Sundance: A rapid prototyping toolkit for parallel PDE simulation and optimization. In L. T. Biegler, O. Ghattas, M. Heinkenschloss, and B. van Bloemen Waanders, editors, *Large-Scale PDE-Constrained Optimization*, Lecture Notes in Computational Science and Engineering. Springer-Verlag, Heidelberg, 2003. Under review by LNCSE series editors.
- [9] K. Long and M. A. Heroux. The Trilinos Solver Framework. In O. Marques and T. Drummond, editors, *2002 ACTS Workshop Proceedings, Berkeley, CA, Sept 3–6, 2002*, 2002.
- [10] F. Natterer. Algorithms in tomography. In I.S. Duff and G.A. Watson, editors, *The State of the Art in Numerical Analysis*. Clarendon Press, 1997.
- [11] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer-Verlag, 1999.
- [12] J.R. Shewchuk. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In M.C. Lin and D. Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996.
- [13] C.R. Vogel. *Computational Methods for Inverse Problems*. SIAM, 2002.