Fast inverse solver for identifying the diffusion coefficient in time-dependent problems using noisy data

Fast inverse solver for identifying the diffusion coefficient in time-dependent problems using noisy data

Jinhui Jiang[1], M Shadi Mohamed[2,*], Mohammed Seaid[3], Hongqiu Li[4]

[1] *State Key Laboratory of Mechanics and Control of Mechanical Structures, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China*

[2] *School of Energy, Geoscience, Infrastructure and Society, Heriot-Watt University, Edinburgh EH14 4AS, UK*

[3] *Department of Engineering, University of Durham, South Road, Durham DH1 3LE, UK*

[4] *Mechatronic Engineering College, Jinling Institute of Technology, Nanjing, 211169, China*

Abstract

We propose an efficient inverse solver for identifying the diffusion coefficient based on few random measurements which can be contaminated with noise. We focus mainly on problems involving solutions with steep heat gradients common with sudden changes in the temperature. Such steep gradients can be a major challenge for numerical solutions of the forward problem as they may involve intensive computations especially in the time domain. This intensity can easily render the computations prohibitive for the inverse problems that requires many repetitions of the forward solution. Compared to the literature we propose to make such computations feasible by developing an iterative approach that is based on the partition of unity finite element method, hence, significantly reducing the computations intensity. The proposed approach inherits the flexibility of the finite element method in dealing with complicated geometries, which otherwise cannot be achieved using analytical solvers. The algorithm is evaluated using several test cases. The results show that the approach is robust and highly efficient even when the input data is contaminated with noise.

Keywords: Inverse problem; Finite element method; Partition of unity method; Diffusion coefficient identification; Transient heat transfer

1 Introduction

The diffusion equation in its general form is one of the fundamental equations in science and engineering. It is often used to describe the behaviour of heat and particles or organisms as they diffuse or spread into a medium [1]. Its applications can be found in a diverse range of topics such as chemically reacting flows [2], molecular peptide structures [3] and subsurface light transport [4], among many others. A key parameter in this equation is the diffusion coefficient which describes the rate at which the heat, particles or organisms spread into the medium. In a given time interval $[0, T]$ and in an open bounded domain $\Omega \subset \mathbb{R}^2$ with a boundary Γ, the linear form of the diffusion equation can be written as

$$\frac{\partial u(t,\mathbf{x})}{\partial t} - D\nabla^2 u(t,\mathbf{x}) = f(t,\mathbf{x}), \qquad (t,\mathbf{x}) \in [0,T] \times \Omega. \tag{1}$$

*Corresponding author: m.s.mohamed@hw.ac.uk

We consider the equation with a mixed-type boundary condition

$$\alpha u + \frac{\partial u}{\partial \mathbf{n}} = g(t, \mathbf{x}), \qquad (t, \mathbf{x}) \in [0, T] \times \Gamma, \tag{2}$$

and a given initial condition

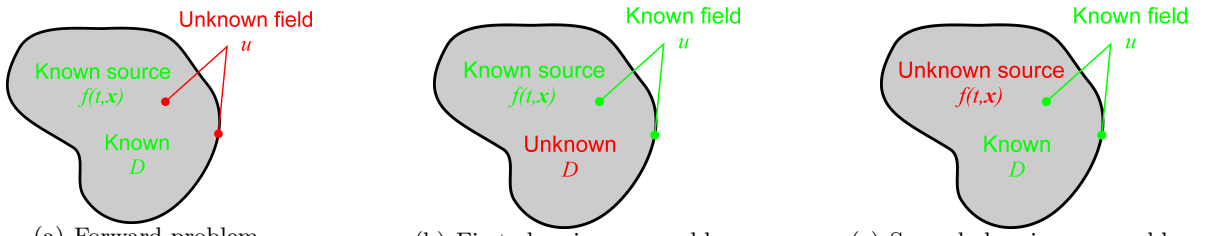
$$u(0, \mathbf{x}) = u_0(\mathbf{x}), \qquad \mathbf{x} \in \Omega. \tag{3}$$

Here, the variables $\mathbf{x} = (x, y)^\top$ denote the spatial Cartesian coordinates, t time and $\mathbf{n}$ the outward unit normal on the boundary while $g(t, \mathbf{x})$ and $u_0(\mathbf{x})$ are some known functions.

The parabolic partial differential equation (PDE) (1) is a quantitative relation between a source/sink or multiple of them described by $f(t, \mathbf{x})$ and the considered field $u(t, \mathbf{x})$. The effect of the medium on the field is described by the diffusion coefficient D such as a smaller coefficient D will cause a slower diffusion rate. In these cases the field may diffuse into the medium at a much smaller rate than the sources/sinks may introduce/absorb the field. This behaviour leads to steep layers in the field, which can be very challenging to resolve numerically. Our main focus is on solving the inverse diffusion problem when such behaviour is observed. The instability in inverse problems, added to the above numerical challenge can cause series difficulties in solving this type of problems.

The inverse solution of the diffusion equations (1)-(3) is structured into two classes of problems [5, 6]. Based on the field measurements collected at accessible parts of the domain, one class identifies the diffusion parameter or in general the subsurface material properties. The second class recovers the unknown sources/sinks or the unknown boundary conditions in the problem. Figure 1 shows the two different classes of the inverse problem alongside the forward problem on a schematic diagram of a given domain. In this study, we solve the inverse problem to identify the diffusion coefficient D. As an application we are interested in heat transfer problems where the coefficient D is the conductivity of the domain. Identifying the coefficient D in this case can be helpful in determining the domain material properties. To solve the forward problem some analytical approaches were established in the parabolic space. The integral transformation method of Fourier [7] as well as of Laplace method [8] and the variable separation method [9] are some examples. However, such analytical methods are in general restricted to domains of simple geometries, hence, are not practical for many engineering applications [10]. To overcome these restrictions a lot of efforts have been invested in developing efficient numerical methods.

Solving the inverse problem of identifying the diffusion coefficient D with numerical methods can be achieved by solving the forward problem and assuming a certain value for the coefficient. This assumption can then be improved iteratively based on a given criterion and re-solving the problem until an accurate estimate is found. Several numerical methods are available in the literature for solving the forward problem [11–14]. A major approach in this regards is the Finite Element Method (FEM). The method offers a remarkable level of accuracy and robustness for complex potential problems governed by steady-state PDEs. A more efficient variation of the FEM was introduced in [15]. The Partition of Unity Method (PUM) involves injecting enrichment functions into the finite element approximation space. The enrichment functions mimic the physical behaviour of the field and hence can better approximate the solution. A key advantage is related to the fact that the enrichment does not disturb inter-element continuity, which makes the approach implementation in an existing finite element code, relatively simple. The Partition of Unity Finite Element Method (PUFEM) was used to solve the Helmholtz equation [16], the diffusion equation [17] and the wave equation [18]. The method was also successful used for solving the inverse Helmholtz problem [19]. In heat transfer problems, using the PUM time-dependent enrichment was implemented in the generalized finite element method to capture sharp thermal gradients [20,21]. The gradients were captured with relatively coarse meshes. However, the time-dependency of the enrichment made it necessary to update the discretization at every time step. To avoid this, time independent enrichments were also introduced [22, 23]. Using the same discretization for all time steps would significantly reduce the computational costs of the



(a) Forward problem. (b) First class inverse problem. (c) Second class inverse problem.

Figure 1: Illustration of different cases for forward and inverse problems of the diffusion equation.

problem. The method was further developed for heterogeneous and non-linear problems [24, 25]. An alternative enrichment approach was achieved by incorporating enrichment functions into the solution approximation space using discontinuous Galerkin methods [26–28].

Inverse diffusion problems attracted a lot of attention starting from the early 1980s. Some of the earlier work on identifying the diffusion coefficient in the steady-state diffusion can be found in [29], while for the nonlinear diffusion equation in [30]. Since then a lot of efforts went into this class of inverse problems. These efforts span a wide range of applications such as recovering the moisture transfer coefficient in a porous material [31] where the finite difference method was used together with the Newton method to minimize an error function. To calculate the diffusion coefficient during quenching process the finite-element, advanceretreat and golden section methods are implemented to solve the inverse heat diffusion equation [32]. Another example is related to identifying the optical properties of a human tissue by measuring the output flux on the tissue outer surface [33]. The work utilized a least-square optimization algorithm and the finite element method. Also coupling the least-square method but with the finite difference method it was possible to determine the unknown diffusion coefficient in [34]. Using measurements taken from a bouncing ball, are used to evaluate the coefficient of restitution which then help to estimate the homogeneity and degree of plasticity of a given material [35]. The material parameter for steady-state reaction-diffusion equations, was estimated on a two-dimensional perforated domain in [36] while a spatially varying potential term was recovered for time-fractional diffusion equations in one-dimension using a quasi-Newton type method [37]. An iterative method based on proper solution space was also employed to evaluate the diffusion coefficient in a two-dimensional elliptic problem [38]. The fixed point theorem and data measured at an interior point are implemented to solve a one-dimensional nonlinear inverse problem to identify a time-dependent convection coefficient in a time-fractional diffusion equation [39].

Compared to the first class, the work on the second class started more recently. In [40] considered the method of fundamental solutions to detect the spatial source term in a multi-dimensional linear diffusion problem with constant and known thermal conductivity. Whereas for spatial fractional anomalous diffusion equation ill-posed inverse problems are first transformed into a well-posed one by a Tikhonov regularization [41]. Thereafter, an algorithm was proposed based on the best perturbation method to estimate the source term. In addition, to deal with ill-posed inverse source problems, a Fourier truncation method was introduced in [42]. All the previous works assume a stationary source term. Indeed, there are only limited number of resources on solving the inverse source problem when the unknown source term is time-dependent [43]. Recently, a technique was developed to identify a time-dependent source term in a time-fractional diffusion equation using the trigonometric method associated with the truncated expansion method [43]. This technique is based on data measured at a fixed time. One of the few examples available on identifying the unknown boundary condition can be found in [44] where an algorithm based on space marching mollification techniques was introduced. An iterative reconstruction algorithm was utilized to recover an unknown source from external boundary measurements in a fractional diffusion problem [45]. To estimate an unknown source location, a two-stage ensemble Kalman filter based on multiscale model reduction was investigated.

In general, one may conclude that the solution of the inverse diffusion problem has been an active

area of research especially in the past few years. However, compared to the forward solution, the progress was relatively slow due to complicated models, incomplete data, intensive computations as well as ill-conditioning and ill-posedness issues [46, 47]. In the present work, we investigate the inverse diffusion problem when the source is known. Based on measurement collected at accessible parts of a considered domain, we aim to identify the diffusion coefficient. The key applications for the proposed algorithm are related to determining the domain material properties. More specifically we are interested in medical applications of microwave and ultrasound for therapeutic purposes where it is highly desirable to identify the material properties by using heat measurements on the human body in order to treat and assess an organ condition [48, 49]. In many cases it is difficult to evaluate the heat diffusion patterns inside a considered object although the heat source is known. Having an accurate tool to evaluate the diffusion coefficient without penetrating the domain would be very useful in this regard. To achieve this it is often necessary to evaluate the inverse solution by going through many repetitions of the forward problem. However, the forward problem solution with standard numerical methods such as the FEM can be computational intensive especially where the solution exhibits steep gradients. Thus, the solution of the inverse problems with numerical methods can become prohibitively expensive in term of the required computations. In this paper we develop an iterative approach based on the partition of unity finite element method which significantly reduces the computations intensity. Hence, the approach enables solving the inverse problem with numerical methods in a wide range of applications that would be otherwise infeasible. The current work is the first to propose an enriched solver for the inverse problem for recovering the diffusion coefficient in time-dependent problems. The rest of the paper is organized as follows. First we present the inverse method to estimate the diffusion coefficient based on the PUM and secant/descent methods. Next, in section 3 we provide some simulations results to study the efficiency and the accuracy of the proposed algorithm. We conclude in section 4 with some remarks and suggestions for future work.

2 Recovering the diffusion coefficient using the PUFEM

We aim to estimate the diffusion coefficient D in the equation (1) based on measurements of the field $u(t, \mathbf{x})$ in some parts of the considered domain Ω. We assume the source $f(t, \mathbf{x})$ and boundary function $g(t, \mathbf{x})$ are known. We approach the problem on two stages. First, we need to build a relation between the diffusion coefficient and the measured data through an inverse modelling scheme. Next, we use an iterative algorithm to recover the unknown diffusion coefficient.

2.1 Forward problem solution

Our inverse modelling scheme is based on the PUM using unstructured grids. To build the scheme we first discretize the boundary-value problem defined by (1)-(3) in time and then in space. The problem time domain is divided into N_t time intervals $[t_n, t_{n+1}]$ of a uniform length Δt where $\Delta t = t_{n+1} - t_n$ for $n = 0, 1, ..., N_t$. To denote the current time step at t_n a super script is added to all time-dependent function *e.g.* u^n. Using a first-order backward Euler time integration scheme, we can rewrite the equation (1) as

$$\frac{u^{n+1} - u^n}{\Delta t} - D\nabla^2 u^{n+1} = f^{n+1}. \tag{4}$$

The time integration is unconditionally stable and the choice of Δt is based only on the targeted accuracy. By rearranging the previous equation we arrive at

$$u^{n+1} - D\Delta t \nabla^2 u^{n+1} = F^{n+1}, \tag{5}$$

where F^{n+1} is given by

$$F^{n+1} = \Delta t f^{n+1} + u^n. \tag{6}$$

Next equation (5) is discretized in space using a standard Galerkin finite element formulation. We first multiply with a trial function W and then integrate over Ω as

$$\int_\Omega W u^{n+1} d\Omega - \int_\Omega D\Delta t W \nabla^2 u^{n+1} d\Omega = \int_\Omega W F^{n+1} d\Omega, \qquad \forall\, W \in H^1(\Omega), \tag{7}$$

with $H^1(\Omega)$ being the conventional Sobolev space. Applying the divergence theorem, (7) reduces to

$$\int_\Omega (D\Delta t \nabla W \cdot \nabla u^{n+1} + W u^{n+1}) d\Omega - \int_\Gamma D\Delta t W \nabla u^{n+1} \cdot \mathbf{n} d\Gamma = \int_\Omega W F^{n+1} d\Omega. \tag{8}$$

Finally, substituting the boundary condition (2) into (8) we obtain the problem weak formulation

$$\int_\Omega (D\Delta t \nabla W \cdot \nabla u^{n+1} + W u^{n+1}) d\Omega + \oint_\Gamma D\Delta t W(\alpha u^{n+1} - g^{n+1}) d\Gamma = \int_\Omega W F^{n+1} d\Omega. \tag{9}$$

The computational domain Ω is discretized into N_e elements. Thus, the weak formulation (9) can be rewritten as a summation over individual elements Ω_e

$$\sum_{e=1}^{N_e} \left[\int_{\Omega_e} \Big(D\Delta t \nabla W \cdot \nabla u^{n+1} + W u^{n+1} \Big) d\Omega_e + \oint_{\Gamma_e} D\Delta t W \Big(\alpha u^{n+1} - g^{n+1} \Big) d\Gamma_e - \int_{\Omega_e} W F^{n+1} d\Omega_e \right] = 0.$$

Using polynomial basis function, the field $u^n(\mathbf{x})$ over the discretized domain can be approximated as

$$u^n(\mathbf{x}) \simeq u_e^n(\mathbf{x}) = \sum_{i=1}^{N_n} u_i^n \mathcal{N}_i(\mathbf{x}) \tag{10}$$

where N_n is the total number of nodal points while $\mathcal{N}_i$ are the polynomial basis functions associated to these nodes and u_i^n are the corresponding nodal values. The key idea in the PUM is to include in the approximation space other types of basis functions that have better approximation properties compared to $\mathcal{N}_i$. The new basis functions are defined based on the problem under study. For the diffusion equation, previous works [17, 22] show that using a combination of exponential functions with the conventional polynomial shape function, has superior approximation properties compared to using only polynomials. The approximated field over the element can then be expressed in terms of the new basis functions as

$$u_e^n(\mathbf{x}) = \sum_{i=1}^{N_n} \sum_{q=1}^{Q} a_i^{n,q} \mathcal{N}_i G_q. \tag{11}$$

The exponential functions also referred to as the enrichment functions [17], are defined by

$$G_q = \frac{\exp\left(-\left(\frac{R_0}{C}\right)^q\right) - \exp\left(-\left(\frac{R_C}{C}\right)^q\right)}{1 - \exp\left(-\left(\frac{R_c}{C}\right)^q\right)}, \qquad q = 1, 2, ..., Q, \tag{12}$$

where $R_0 := |\mathbf{x} - \mathbf{x}_c|$ is the distance from the control point $\mathbf{x}_c$ to any point in the domain $\mathbf{x}$. The constant R_c and C control the steepness of the exponential function G_q. Using the new approximation the diffusion equation is to be solved for the values of the new set of unknowns $a_i^{n,q}$. For an element Ω_e we define two vectors $\mathbf{N}_e$ and $\mathbf{a}_e^n$ by

$$\mathbf{N}_e = \Big\{ \mathcal{N}_1 G_1 \quad \mathcal{N}_1 G_2 \quad ... \quad \mathcal{N}_1 G_Q \quad \mathcal{N}_2 G_1 \quad ... \quad \mathcal{N}_2 G_Q \quad ... \quad \mathcal{N}_M G_Q \Big\}^\top,$$
$$\mathbf{a}_e^n = \Big\{ a_1^{n,1} \quad a_1^{n,2} \quad ... \quad a_1^{n,q} \quad a_2^{n,1} \quad ... \quad a_2^{n,q} \quad ... \quad a_M^{1,q} ... \quad a_M^{n,q} \Big\}^\top,$$

with M being the total number of nodes for this element. We also build a rectangular matrix $\mathbf{L}_e$ made of zeros and ones so that the elementary nodal unknowns $\mathbf{a}_e^n$ and the global unknown vector $\mathbf{a}^n$ are related

$$\mathbf{a}_e^n = \mathbf{L}_e \mathbf{a}^n. \tag{13}$$

Similarly, we can also relate other elementary and global quantities as

$$\begin{aligned} \nabla u^n &= \nabla \mathbf{N}_e \cdot \mathbf{L}_e \mathbf{a}^n \\ &= \mathbf{B} \cdot \mathbf{L}_e \mathbf{a}^n \end{aligned} \tag{14}$$

Making use of this definition we can rewrite the matrix form of the weak formulation as

$$\begin{aligned} \sum_{e=1}^{N_e} \Bigg[\mathbf{L}_e^\top \left(\int_{\Omega_e} \left(D\Delta t \mathbf{B}^\top \mathbf{B} + \mathbf{N}_e^\top \mathbf{N}_e \right) d\Omega_e + D\Delta t \alpha \oint_{\Gamma_e} \left(\mathbf{N}_e^\top \mathbf{N}_e \right) d\Gamma_e \right) \mathbf{L}_e \mathbf{a}^{n+1} \\ + \mathbf{L}_e^\top \left(-D\Delta t \oint_{\Gamma_e} \left(\mathbf{N}_e^\top g^{n+1} \right) d\Gamma_e - \oint_{\Omega_e} \left(\mathbf{N}_e^\top F^{n+1} \right) d\Omega_e \right) \Bigg] = 0. \end{aligned} \tag{15}$$

This form can be simplified by separating the contribution of individual integral terms to matrices on elements level as

$$\begin{aligned} \mathbf{K}_e &= \underbrace{D\,\Delta t \int_{\Omega_e} \mathbf{B}^\top \mathbf{B} d\Omega}_{\mathbf{K}_{e,k}} + \underbrace{\int_{\Omega_e} \mathbf{N}_e^\top \mathbf{N}_e d\Omega_e}_{\mathbf{K}_{e,m}} + \underbrace{D\,\alpha \Delta t \oint_{\Gamma_e} \mathbf{N}_e^\top \mathbf{N}_e d\Gamma_e}_{\mathbf{K}_{e,c}}, \\ \mathbf{f}_e^{n+1} &= \underbrace{D\,\Delta t \oint_{\Gamma_e} \mathbf{N}_e^\top g^{n+1} d\Gamma}_{\mathbf{f}_{e,g}^{n+1}} + \underbrace{\oint_{\Omega_e} \mathbf{N}_e^\top F^{n+1} d\Omega}_{\mathbf{f}_{e,f}^{n+1}}. \end{aligned} \tag{16}$$

Hence, we can identify the contribution of the diffusion coefficient to each of these matrices. High order Gaussian quadrature is needed to calculate the integrals because of the exponential basis functions. The element matrices are then assembled into the global system matrix

$$\underbrace{\left[\sum_{e=1}^{N_e} \mathbf{L}_e^\top \mathbf{K}_e \mathbf{L}_e \right]}_{\mathbf{K}} \mathbf{a}^{n+1} - \underbrace{\left\{ \sum_{e=1}^{N_e} \left(\mathbf{L}_e \right)^\top \mathbf{f}_e^{n+1} \right\}}_{\mathbf{f}^{n+1}} = 0. \tag{17}$$

The system matrix $\mathbf{K}$ is independent of time thanks to the time-independence of the enrichment function. To solve this system of equations we find the Moore-Penrose pseudo-inverse $\mathbf{K}^+$ of $\mathbf{K}$. Thus, the solution vector can be obtained as

$$\mathbf{a}^{n+1} = \mathbf{K}^+ \mathbf{f}^{n+1}. \tag{18}$$

Once the nodal unknown vector $\mathbf{a}^{n+1}$ is obtained, the field can be determined inside any element Ω_e based on the equations (11) and (13)

$$\begin{aligned} u^{n+1}(\mathbf{x}_\mathrm{e}) &= \mathbf{N}_\mathrm{e} \cdot \mathbf{L}^\mathrm{e} \mathbf{a}^{n+1} \\ &= \mathbf{N}_\mathrm{e} \cdot \mathbf{L}^\mathrm{e} \left[\sum_{e=1}^{N_e} \mathbf{L}_e^\top \left(D\mathbf{K}_{e,k} + \mathbf{K}_{e,m} + D\mathbf{K}_{e,c} \right) \mathbf{L}_e \right]^+ \left\{ \sum_{e=1}^{N_e} \mathbf{L}_e^\top \left(D\mathbf{f}_{e,g}^{n+1} + \mathbf{f}_{e,f}^{n+1} \right) \right\}, \end{aligned} \tag{19}$$

where $\mathbf{x}_\mathrm{e} \in \Omega_\mathrm{e}$. The equation (19) presents the relationship between the diffusion coefficient and the field solution at a given location $\mathbf{x}_\mathrm{e}$. The relation between the field u^{n+1} and the diffusion coefficient D is severely nonlinear because both $\mathbf{f}$ and $\mathbf{K}^+$ are dependent on the diffusion coefficient. Therefore, in general it is not possible to directly solve the inverse problem of identifying the diffusion coefficient D in term of u^{n+1}. Furthermore, finding an analytical expression is also not possible if the geometry is complicated. In this study, we propose an iterative algorithm to estimate the diffusion coefficient by defining and minimizing an objective function.

2.2 Iterative algorithm to estimate the diffusion coefficient

In this study, we aim to identify the diffusion coefficient based on data measured at accessible parts of the considered domain at some time instants t^{n+1}. Assuming that the data is collected at different locations $\mathbf{x}_j$ with $j = 1, 2, ...N_l$ and stored in a vector $u_{\text{exc}}^{n+1}(\mathbf{x}_j)$. We start by assuming a certain value for the diffusion coefficient $\hat{D}$. Hence, the forward problem is solved based on $\hat{D}$ to reach the time instant t^{n+1}. We define a residual function $f_j(D)$ which is the difference between the measured data $u_{\text{exc}}^{n+1}(\mathbf{x}_j)$ and the numerical prediction $\hat{u}^{n+1}(\mathbf{x}_j)$ at the jth data point and at time t^{n+1}

$$f_j(\hat{D}) = u_{\text{exc}}^{n+1}(\mathbf{x}_j) - \hat{u}^{n+1}(\mathbf{x}_j), \tag{20}$$

with

$$\hat{u}^{n+1}(\mathbf{x}_j) = \mathbf{N}_{\text{e}}(\mathbf{x}_j) \cdot \mathbf{L}^{\text{e}} \left[\sum_{e=1}^{N_e} (\mathbf{L}_e)^\top \left(\hat{D}\mathbf{K}_{e,k} + \mathbf{K}_{e,m} + \hat{D}\mathbf{K}_{e,c} \right) \mathbf{L}_e \right]^+ \left\{ \sum_{e=1}^{N_e} (\mathbf{L}_e)^\top \left(\hat{D}\mathbf{f}_{e,g}^{n+1} + \mathbf{f}_{e,f}^{n+1} \right) \right\}.$$

Here, the element e is chosen because $\mathbf{x}_j \in \Omega_{\text{e}}$. We then define an objective function $\vartheta(\hat{D})$ for all the data points as

$$\vartheta(\hat{D}) = \frac{1}{N_l} \sum_{j=1}^{N_l} \frac{\left| f_j(\hat{D}) \right|}{\left| u_{\text{exc}}^{n+1}(\mathbf{x}_j) \right|} = \frac{1}{N_l} \sum_{j=1}^{N_l} \frac{\left| u_{\text{exc}}^{n+1}(\mathbf{x}_j) - \hat{u}^{n+1}(\mathbf{x}_j) \right|}{\left| u_{\text{exc}}^{n+1}(\mathbf{x}_j) \right|}. \tag{21}$$

Next, we use the secant and the descent methods to improve the estimate $\hat{D}$ iteratively. Based on local convergence theorem [50], the secant method helps to ensure the convergence when the initial value $\hat{D}$ is close to the exact root D. Whereas, the descent method ensures the convergence when the initial guess is far from the exact diffusion coefficient. Let $i = 1, 2, ...$ be the iterative step and $\hat{D}_i$ be the estimate at the ith iteration. The diffusion coefficient is improved using the following equation

$$\hat{D}_{i+1} = \hat{D}_i - \mu f_j(\hat{D}_i) \frac{\hat{D}_i - \hat{D}_{i-1}}{f_j(\hat{D}_i) - f_j(\hat{D}_{i-1})}, \tag{22}$$

where $0 < \mu \leq 1$ is the descent factor. The value of μ is chosen so that the following inequality is met

$$\left| f_j(\hat{D}_{i+1}) \right| < \left| f_j(\hat{D}_i) \right| \tag{23}$$

We initially take $\mu = 1$. If the condition (23) is not met we reduce μ by a factor of 0.5 and repeat this until the condition is satisfied. It should be noted that in order to ensure the direction is descent the descent factor is adjusted. This is achieved by reducing the descent factor until the condition stated in Equation (23) is met. However, if after reducing the descent factor the condition still cannot be met, we have to change the initial value and try again. We also restrict ourselves to the cases where the diffusion coefficient is a positive real number. The value of $\hat{D}_i$ is reset to $\hat{D}_i > 0$ whenever an estimate $\hat{D}_i \leq 0$ is obtained. The detailed steps of the proposed algorithm are listed in Figure 2.

The algorithm involves two nested iterative processes: The inner process estimate the diffusion coefficient $D_i^j, j = 1, 2, ...N$ based on the secant and descent methods. The outer process determine an appropriate number of enrichment functions Q for the PUM solution. Obviously, more enrichment leads to better accuracy but the extra enrichment also increase the computational costs and leads to an ill-conditioned linear system [17]. Therefore, it is important to use enough enrichment functions but not too many of them. The balance between the number of enrichment functions and the conditioning of the linear system is previously discussed in details, see for instance [17]. The inner process is terminated once either of the following conditions is met

$$|\hat{D}_{i+1} - \hat{D}_i| \leq \xi_1, \qquad \text{or} \qquad |f_j(\hat{D}_i)| \leq \xi_1, \tag{24}$$

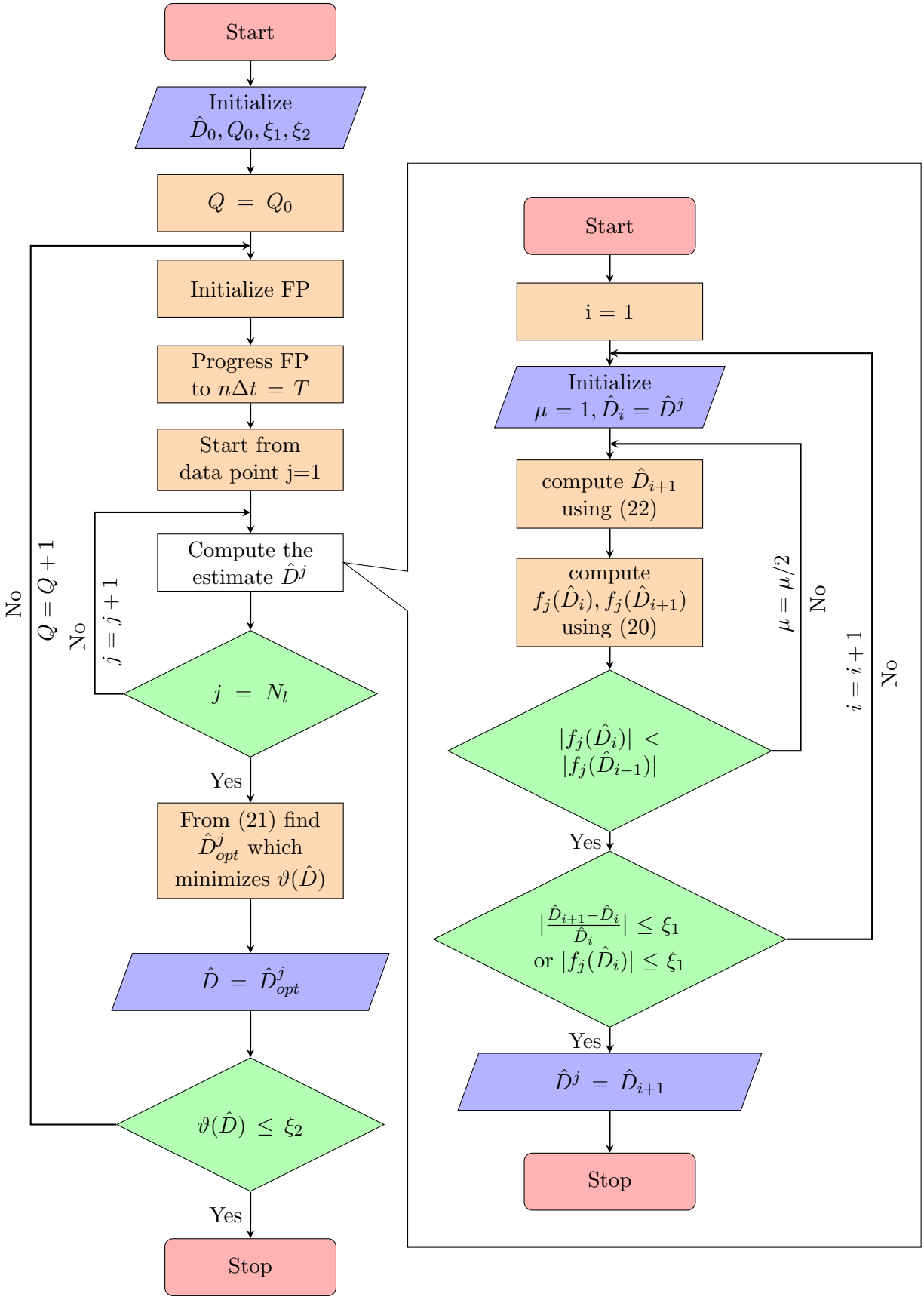


Figure 2: Flowchart of the iterative algorithm used in the current study for recovering the diffusion coefficient. Here, FP refers to the forward problem

while the outer process is terminated when

$$\vartheta(\hat{D}) \leq \xi_2, \tag{25}$$

where ξ_1 and ξ_2 are predefined tolerances. Since, the diffusion coefficient is estimated for each data point, N_l estimates are found *i.e.* D^j with $j = 1, 2, ...N_l$. The estimate D^j which minimizes the objective function $\vartheta(\hat{D})$ is considered as the best estimate $\hat{D}^j_{opt}$. If the condition $\vartheta(\hat{D}^j_{opt}) \leq \xi_2$ is not satisfied then a next round of iteration is started with the value of $\hat{D}_0$ is set to be $\hat{D}^j_{opt}$.

Based on the knowledge of the exact diffusion coefficient the estimation error is calculated in two different methods. To verify the proposed algorithm accuracy, we study cases where the exact diffusion coefficient D is known. Thus, we can use the following error norm

$$\varepsilon_1 = \frac{\left|\hat{D} - D\right|}{D} \times 100\%. \tag{26}$$

For the case with unknown exact diffusion coefficient, we propose utilizing N_v additional data points. The field measured at these points $u^{*,n}(\mathbf{x}_j)$ is used as a verification data to test the accuracy of the estimated result as

$$\varepsilon_2 = \frac{1}{N_v} \sum_{j=1}^{N_v} \frac{|\hat{u}^n(\mathbf{x}_j) - u^{*,n}(\mathbf{x}_j)|}{|u^{*,n}(\mathbf{x}_j)|} \times 100\%, \tag{27}$$

where $u^n(\mathbf{x}_j)$ is the field computed at the verification points using the estimated diffusion coefficient $\hat{D}$. It should be stressed that the proposed algorithm can estimate the diffusion coefficient even if only one data point is known. However, to ensure the convergence of the algorithm to the correct diffusion coefficient it is advised that more data points should be accounted for in the algorithm.

3 Numerical results

To investigate the efficiency and the accuracy of the proposed algorithm two examples are considered. In the first test example a problem with a given analytical solution is considered while in the second no analytical solution is available. In both examples two data points are used for estimating the coefficient D. A third data point will also be used to verify the estimate. The presented finite element solutions are obtained on quadrilateral elements with piecewise bilinear shape functions and the enrichment parameters are given as $R_c = \sqrt{\frac{2800}{239}}$ and $C = \frac{200}{239}$ which are the same values used in [51]. It should be noted that other values of the same order of magnitude were also considered with no significant changes in the presented results. The control point $\mathbf{x}_c$ is located at the center of the computational domain in each example. All the integrals are evaluated using Gauss quadrature. We also ensure that enough integration points are used by increasing the number of integration points until the FEM solution has converged. This is done at the first iteration. The converged number of integration points is then retained for all the remaining iterations.

3.1 Diffusion problem with known exact solution

We consider a problem with the exact solution given by

$$u(x, y, t) = x^{20}y^{20}(2 - x)^{20}(2 - y)^{20}(2 - e^{-D^*t}) \tag{28}$$

where the considered computational domain is $\Omega = [0, 2] \times [0, 2]$. The problem was first studied in [17]. The source term f, the boundary condition g and the initial condition u_0 of the boundary value problem (1)-(3), are all explicitly calculated based on the exact solution (28). The diffusion coefficient is $D^* = 0.01$. Figure 3 shows the exact solution at $t = 0.5$.

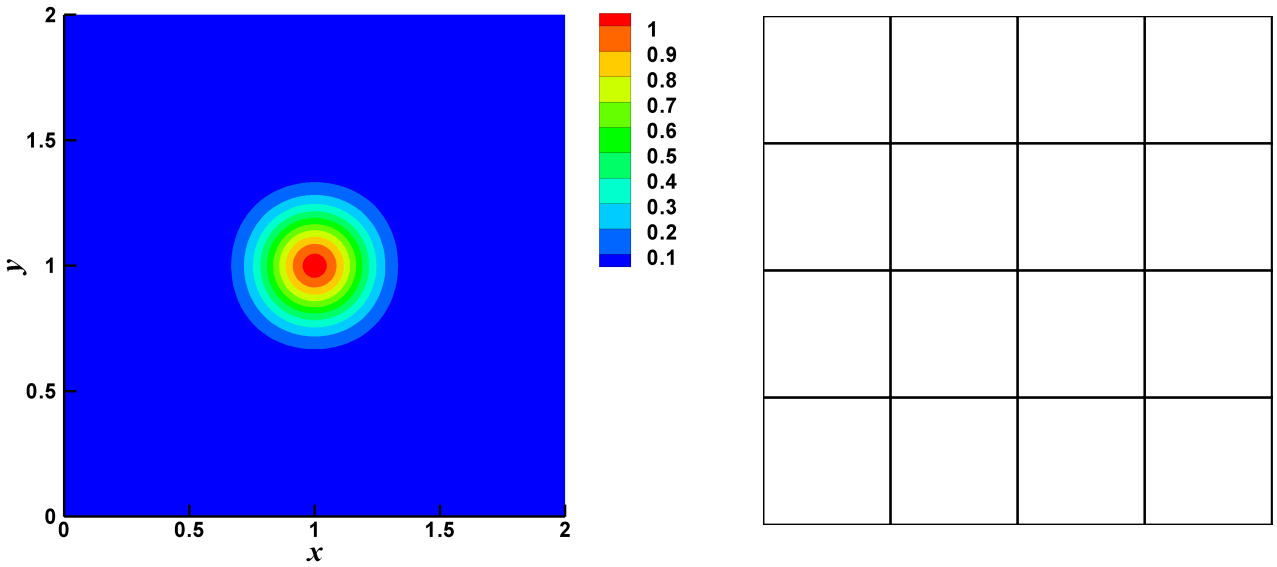


Figure 3: Problem 1: Analytical solution for $D^* = 0.01$ (left) and the considered mesh (right).

To estimate the diffusion coefficient the forward problem is solved on a uniform mesh composed of 16 square 4-noded elements and 25 nodes shown in Figure 3. It should be noted that this problem involves steep gradients which require refined FEM mesh to recover. However, thanks to the enrichment it is possible to use a coarse mesh which significantly reduced the computational costs in this problem [17]. The initial guess is $D_0 = 0.006$ and the initial number of the enrichment functions is $Q_0 = 4$. The solution is evaluated at $t = 0.2$ and the timestep is $\Delta t = 0.01$. The considered data points are $P_1 = (1.0, 0.75)$ and $P_2 = (1.0, 1.0)$ with the control point being $P_3 = (0.5, 1)$. Table 1 shows the iterations performed to improve the estimated diffusion coefficient. The considered stopping tolerances are $\xi_1 = 10^{-5}$ and $\xi_2 = 10^{-5}$. The final estimate is $\hat{D}^q_{opt} = 0.01001$ and is achieved at P_2 with $Q = 7$. The corresponding value for the objective function at this point is $\vartheta(\hat{D}) = 1.26 \times 10^{-6}$ which is slightly lower than the value of the function at P_1. The corresponding values at P_1 are $\hat{D}^q_{opt} = 0.00999$ and $\vartheta(\hat{D}) = 2.27 \times 10^{-6}$. The error in the converged estimate of the diffusion coefficient is $\varepsilon_1 = 9.02 \times 10^{-2}\%$. The exact solution at P_3 is 0.0058953 whereas the value of the numerical solution is 0.0058974 which is obtained from the converged estimate. Thus, the value of the error at the control point P_3 is $\varepsilon_2 = 3.61 \times 10^{-2}\%$.

As the forward problem progresses in time, the solution accumulates temporal errors of the time stepping scheme. Our second aim in this example is to evaluate the stability of the proposed algorithm for measurements taken at different time instants. Because in this example the diffusion coefficient is invariable in time the algorithm should converge to the same estimate from measurements taken at any time instant. Here, we consider three time instants, namely, $t = 0.2, 0.5$ and 2.0 using the timestep size $\Delta t = 0.05$. At these time instants the solution is evaluated in $P_1 = (1, 0.25)$ and $P_2 = (0.5, 1.75)$ as well as the control point $P_3 = (0.5, 1)$ to evaluate the final error. The stopping tolerances are $\xi_1 = 10^{-4}$ and $\xi_2 = 10^{-4}$ in this case. These tolerances are slightly increased from the previous study to reflect the higher solution errors expected here as we consider larger timestep size *i.e.* $\Delta t = 0.05$ as well as later time instants *i.e.* $t = 0.5$ and 2.0 compare to before.

Again the forward problem is solved on the coarse mesh shown in Figure 3. For the cases considered here the initial number of enrichment functions is taken $Q_0 = 7$. This is based on the results obtained in the first set of results where the best estimate in this example was achieved at this number of enrichment functions. Again we start with the initial guess $D_0 = 0.006$. Table 2 shows the iterations required to converge to the best estimate of the diffusion coefficient for the three considered time instants. The results show that the number of iterations increases as we consider a later point in time.

Table 1: Example 1: Iterations for $Q = 4, 5, 6$ and 7 with $\Delta t = 0.01$ and at $t = 0.2$.

# Itr	$Q = 4$ $(\hat{D}_0 = 0.006)$		$Q = 5$ $(\hat{D}_0 = 0.0126)$		$Q = 6$ $(\hat{D}_0 = 0.00812)$		$Q = 7$ $(\hat{D}_0 = 0.01860)$	
	P_1	P_2	P_1	P_2	P_1	P_2	P_1	P_2
1	0.0066	0.0066	0.0139	0.0139	0.0089	0.0089	0.0205	0.0205
2	0.0200	0.0117	0.0249	0.0056	0.0020	0.0155	0.0089	0.0073
3	0.0240	0.0125	0.0309	0.0032	0.0325	0.0178	0.0072	0.0047
4	0.0237	0.0126	0.0325	0.0078	0.0321	0.0186	0.0099	0.0095
5	0.0234			0.00808	0.0317	0.0186	0.00999	0.0099
6	0.0231			0.00812	0.0313	0.0165		0.00998
7	0.0228				0.0309	0.01864		0.01001
8	0.0225				0.0304	0.01860		
9	0.0222				0.0300			
10	0.0219				0.0296			
20	0.0186				0.0252			
40	0.0126				0.0171			
50	0.0096				0.0130			
$\hat{D}^q_{opt}$	0.0096	0.0126*	0.0325	0.00812*	0.0130	0.01860*	0.00999	<u>0.01001</u>
$\vartheta(\hat{D})$	0.0064	0.0045	0.0067	0.0039	0.0039	0.0032	2.27×10^{-6}	1.26×10^{-6}

* value used for the next round of iterations.

Table 2: Example 1: Iterations for $Q = 7$, $D_0 = 0.006$ and $\Delta t = 0.05$ at different time instants.

# Itr	$t = 0.2$		$t = 0.5$		$t = 2.0$	
	P_1	P_2	P_1	P_2	P_1	P_2
1	0.006600	0.006600	0.006600	0.006600	0.006600	0.006600
2	0.008426	0.001500	0.026834	0.009829	0.009075	0.009016
3	0.009384	0.008719	0.002775	0.009980	0.009803	0.009806
4	0.009864	0.010116	0.000353	0.009997	0.009989	0.009990
5	0.009987	0.010287	0.002901	0.009999	0.009999	0.009995
6		0.009996	0.002988	0.010007	0.009974	0.009974
7		0.009992	0.003198	0.009998	0.010022	0.010013
8			0.003212	0.009997	0.010032	0.010390
9					0.010018	0.009998
$\hat{D}^q_{opt}$	0.009987	<u>0.009992</u>	0.003212	<u>0.009997</u>	0.010018	<u>0.009998</u>
$\vartheta(\hat{D})$	2.30×10^{-04}	8.27×10^{-05}	$1.45 \times 10^{+00}$	4.29×10^{-04}	7.79×10^{-04}	7.70×10^{-05}

Table 3: Example 1: Iterations for $Q = 7$, $D_0 = 0.006$ and $\Delta t = 0.5$ at different time instants.

# Itr	0% noise		5% noise	
	P_1	P_2	P_1	P_2
1	0.006600	0.006600	0.006600	0.006600
2	0.018109	0.009794	0.015941	0.009786
3	0.001385	0.009985	0.002308	0.009986
4	0.024000	0.010001	0.001500	0.010005
5	0.001500		0.002761	0.010010
6	0.023400		0.002968	
7	0.022800		0.003158	
8	0.005556		0.003180	
9	0.000435			
10	0.006217			
15	0.009776			
16	0.009957			
$\hat{D}^q_{opt}$	0.009957	<u>0.010001</u>	0.003180	<u>0.010010</u>
$\vartheta(\hat{D})$	8.16×10^{-04}	4.05×10^{-04}	1.81×10^{00}	6.80×10^{-03}

For example the estimate based on P_2 data, converges after 9 iterations at $t = 2.0$ whereas for the same data points but at $t = 0.2$ the estimate converges after 7 iterations. This might be attributed to the increased forward problem temporal error as the solution progresses in time. The final errors corresponding to the results listed in Table 2 are $\varepsilon_1 = 7.68 \times 10^{-4}\%, 3.04 \times 10^{-4}\%$ and $1.55 \times 10^{-4}\%$ and $\varepsilon_2 = 5.62 \times 10^{-3}\%, 1.30 \times 10^{-2}\%$ and $3.33 \times 10^{-2}\%$ for $t = 0.2, 0.5$ and 2.0, respectively.

Our final aim in this example is to understand the impact of data contamination on the stability of the algorithm. To this end the data measured in $P1 = (1, 0.25)$ and $P_2 = (0.5, 1.75)$ at $t = 0.5$ is contaminated with a zero-mean white Gaussian noise with the noise-to-signal ratio of 5%. The problem is solved again on the same mesh as before with $Q_0 = 7$ with $\Delta t = 0.01$ for both the contaminated and the clean data. Using very small tolerances can be unrealistic in this study as we know that the data is contaminated with 5% noise. Therefore we consider higher tolerances than before $\xi_1 = 10 \times 10^{-3}$ and $\xi_2 = 10 \times 10^{-2}$ to reflect the uncertainty in the input data. Using smaller tolerances will not improve the estimate accuracy but will only increase the number of iterations required for convergence. The control data point is again $P_3 = (0.5, 1)$. We start again with the same initial guess as before $D_0 = 0.006$. Table 3 shows the performed iterations and the estimated diffusion coefficients at each iteration for clean as well as contaminated data. Similar to the previous study, the estimate converges first at P_2 with 4 iterations for the clean data and 5 for the contaminated. At P_1 the estimate converges after 16 iterations and 8 iterations for the clean and contaminated data, respectively. In general one may conclude that the algorithm is stable even when contaminated data is used. However, the case of the contaminated data in P_1 seems to suffer from a slow convergence rate. The estimate $\hat{D}^8 = 0.003180$ in this case is much smaller than the actual diffusion coefficient. This is also indicated by the value of the objective function which is relatively large $\vartheta(\hat{D}) = 1.81$. Obviously, it is possible to improve this by starting another set of iterations with an increased Q. However, because the objective function value at P_2 is already smaller than the tolerance ξ_2 we consider the estimate at P_2 to be the best estimate.

3.2 Diffusion problem with unknown exact solution

The second example studies a circular domain with a rectangular heat source. The domain is of a unity radius $r = 1$ with the heat source located at the centre. The heat source is assumed to be

activated inside the domain for a given time span and is defined by

$$f(t,\mathbf{x}) = \begin{cases} 1800\ K/s, & \text{if} \quad (x,y) \in [0.8,0.9] \times [1.2,1.1] \qquad \text{and} \qquad t \leq 5\ s, \\ 300\ K/s, & \text{elsewhere.} \end{cases}$$

Figure 4 shows a schematic plot of the computational domain and the source. The initial domain temperature is given as $u_0 = 300$. The domain ambient temperature is assumed fix so that $g = 300$.

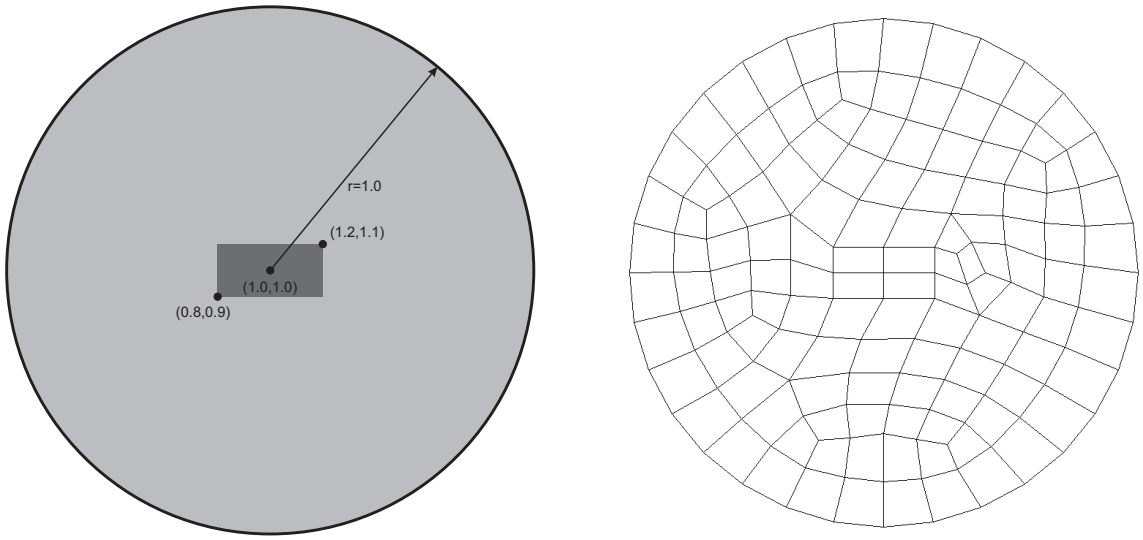


Figure 4: Problem 2: schematic plot of the computational domain (left), the considered mesh (right).

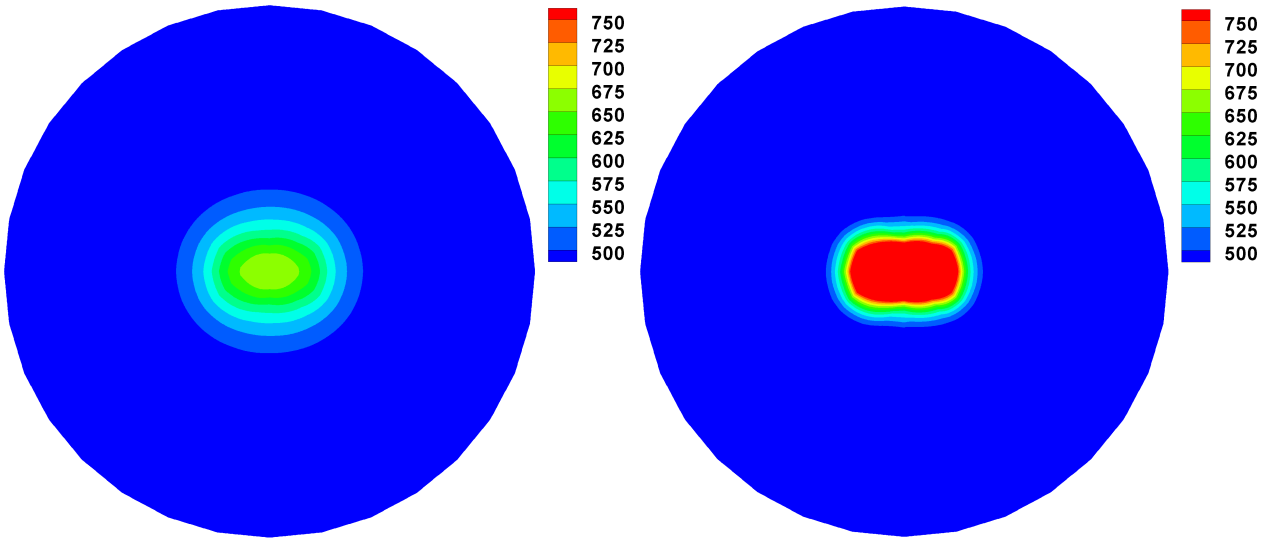


Figure 5: Problem 2: the reference solution at $t = 0.5$ and for $D = 0.1$(left) and $D = 0.01$ (right).

Two values for the diffusion coefficient of the domain material are studied in this example. In the first instant we consider $D = 0.1$. Three different points, namely, $P_1 = (1.16667, 1.08883), P_2 = (0.66667, 0.08883)$ and $P_3 = (0.75, 1.125)$ are considered for the input data. The points are chosen so that P_1 is inside the heat source close to the domain core, P_2 is very close to the domain boundary while P_3 is located outside the source but away from the boundary. We simulate the problem heat transfer using the standard finite element method so that a converged reference solution is obtained on a highly refined mesh. This reference solution is then used to evaluate the domain temperature at

Table 4: Example 2: Iterations for $Q = 4, 5, 6, 7$ and 8 with $\Delta t = 0.1$ and at $t = 0.5$.

# Itr	$Q=4$ $(\hat{D}_0 = 0.02)$		$Q=5$ $(\hat{D}_0 = 0.006754)$		$Q=6$ $(\hat{D}_0 = 0.009526)$		$Q=7$ $(\hat{D}_0 = 0.092051)$		$Q=8$ $(\hat{D}_0 = 0.118309)$	
	P_1	P_2	P_1	P_2	P_1	P_2	P_1	P_2	P_1	P_2
1	0.0220	0.022	0.007430	0.007430	0.010479	0.010479	0.101256	0.101256	0.13014	0.13014
2	0.0030	0.002	0.025273	0.009098	0.048988	0.028373	0.114575	0.609097	0.09864	0.097793
3	0.0020	0.200	0.031947	0.009479	0.076578	0.040725	0.117913	0.009205	0.096809	0.092782
4	0.006138	0.198	0.034595	0.009525	0.095262	0.056504	0.118299	0.920512	0.099958	0.100812
5	0.00664	0.196	0.034917	0.009526	0.094309	0.071008	0.118309	0.911307	0.100008	0.101235
6	0.006754	0.194	0.034928		0.093357	0.083113		0.902102	0.100017	0.099794
7		0.192			0.092404	0.089778		0.892897		
8		0.190			0.091451	0.091805		0.883692		
9		0.188			0.090499	0.092044		0.874487		
10		0.186			0.089546	0.092051		0.865282		
15		0.174			0.083830			0.810051		
20		0.164			0.079067			0.764025		
30		0.144			0.069541			0.671974		
40		0.124			0.060015			0.579923		
50		0.104			0.050489			0.487872		
$\hat{D}^q_{opt}$	0.006754^*	0.104	0.034928	0.009526^*	0.050489	0.092051^*	0.118309^*	0.487872	<u>0.100017</u>	0.099794
$\vartheta(\hat{D})$	0.1652	0.4616	0.1105	0.0015	0.0248	4.96×10^{-3}	0.1588	0.1656	1.10×10^{-4}	4.38×10^{-4}

* value used for the next round of iterations.

Table 5: Example 2: Iterations for $Q = 4, 5, 6, 7$ and 8 with $\Delta t = 0.1$ and at $t = 0.5$.

# Itr	$Q=4$ $(\hat{D}_0 = 0.02)$		$Q=5$ $(\hat{D}_0 = 0.007237)$		$Q=6$ $(\hat{D}_0 = 0.009531)$		$Q=7$ $(\hat{D}_0 = 0.092208)$		$Q=8$ $(\hat{D}_0 = 0.124306)$	
	P_1	P_2	P_1	P_2	P_1	P_2	P_1	P_2	P_1	P_2
1	0.0220	0.022	0.007961	0.007961	0.010484	0.010484	0.101429	0.101429	0.136737	0.136737
2	0.0039	0.002	0.026374	0.009261	0.049644	0.028384	0.118674	0.611404	0.103284	0.097385
3	0.0020	0.200	0.033180	0.009509	0.078151	0.040743	0.123534	0.009221	0.101199	0.092222
4	0.006591	0.198	0.035936	0.009531	0.095309	0.056538	0.124279	0.922085	0.104761	0.100136
5	0.007109	0.196	0.036276	0.009531	0.094356	0.071071	0.124306	0.912864	0.104811	0.100632
6	0.007234	0.194	0.036288		0.093403	0.083215		0.903643	0.104811	0.100711
7	0.007237	0.192	0.036288		0.092450	0.089916		0.894422		0.100552
8		0.190			0.091496	0.091958		0.885202		0.100928
9		0.188			0.090543	0.092200		0.875981		0.100288
10		0.186			0.089592	0.092208		0.866760		0.100041
15		0.174			0.083872			0.820656		0.100336
20		0.164			0.079106			0.765331		0.100413
30		0.144			0.069575			0.673122		
40		0.124			0.060045			0.580913		
50		0.104			0.050514			0.488705		
$\hat{D}^q_{opt}$	0.007237^*	0.104	0.036288	0.009531^*	0.050514	0.092208^*	0.124306^*	0.488705	0.104811	<u>0.100413</u>
$\vartheta(\hat{D})$	0.1729	0.4615	0.1128	0.0016	0.0249	5.51×10^{-3}	0.1577	0.1657	9.83×10^{-4}	1.41×10^{-4}

* value used for the next round of iterations.

P_1, P_2 and P_3 at the time instant $t = 0.5$. Next, 5% Gaussian noise is added to the reference solution. Another set of input data is then generated using the reference solution contaminated with the noise.

In the following simulations the P_1 and P_2 are used as data points while P_3 is used as a control point. The initial value considered at the first iteration are $D_0 = 0.02$ for the diffusion coefficient, $Q = 4$ for the enrichment functions, $\xi_1 = 5 \times 10^{-04}$ and $\xi_2 = 10^{-05}$ for the stopping tolerances. It should be stressed that the initial estimate of the diffusion coefficient is chosen to be one order of magnitude smaller than the exact value. The aim is to evaluate if the proposed algorithm can still converge if no accurate estimates are available. The forward problem is solved on a coarse finite element mesh composed of 123 4-noded elements and 140 nodes as shown in Figure 4. The considered timestep size is $\Delta t = 0.1$. Table 4 summarizes the iterations needed before the algorithm converges to an accurate estimate if the clean data is used. The results show that the convergence may require up to 50 iterations at either of the data points for $Q = 4, 5, 6, 7$ and 8. Furthermore, the estimate diverges rather than converging until we increase the number of enrichment functions to $Q = 6$, where the estimate starts to converge. This can be related to the finite element approximation as using a small number of enrichment functions can only provide a rough approximation. Using six or more enrichment functions seems to be suitable in this case. However, the algorithm still shows stable behaviour and keep increasing the number of enrichment functions leads to the convergence eventually. Figure 5 shows the converged solution of the problem. We repeated the same study and starting with the same initial values as before but this time using the data contaminated with noise. The output at different iterations is summarized in Table 5. In general one can see consistency between the performance of the algorithm for both contaminated and clean data. This can be seen in the number of iterations at each point in all the loops of increasing the number of enrichment functions. It can also be seen in the best estimate achieved in each loop of iterations. For example at the $Q = 4$ loop the algorithm requires 6 and 50 iterations for P_1 and P_2, respectively, for the clean data. The algorithm converges to $D^j_{opt} = 0.006754$ in this case. The respective numbers for the contaminated data are: 7 and 50 iterations and $D^j_{opt} = 0.007237$. This consistency suggests that the algorithm can be robust even when the input data involves some uncertainty. The errors at the control point are $\varepsilon_2 = 9.62 \times 10^{-2}\%$ and $\varepsilon_2 = 9.79 \times 10^{-2}\%$, respectively. The final estimate errors for clean and contaminated data cases are $\varepsilon_1 = 1.7 \times 10^{-3}\%$ and $\varepsilon_1 = 4.1 \times 10^{-3}\%$, respectively. This again shows that the algorithm is consistent even when the input data include some noise.

The second value considered for the diffusion coefficient is $D = 0.01$. The smaller value of the diffusion coefficient in this part of the calculations results in steeper heat gradients. To recover such gradients it is necessary to refine the finite element mesh around them. We followed the same procedure as before. Again three points are considered inside the domain $P_1 = (0.908748, 0.820353), P_2 = (0.986766, 0.513581)$ and $P_3 = (0.75, 1.125)$. In this set of results all the points are chosen in the vicinity of the heat source but not inside it. As in the previous set of results the FEM is used to create a reference solution where two sets of data are considered *i.e.* clean and contaminated with 5% Gaussian noise.

The same initial values as before are used again. However, we use now $Q = 8$ as an initial value for the enrichment. This number is based on the previous set of results in order to avoid unnecessary iterations. The initial guess for the diffusion coefficient is again taken to be one order of magnitude smaller than the exact one *i.e.* $D^0_{opt} = 0.002$. The forward problem is solved on the same mesh and using the same timestep size as before. Table 6 summarizes the output of the required iterations for clean and contaminated data to achieve a converged. The algorithm converges within 10 iterations at P_1 and P_2 for clean and contaminated data. The convergence is again consistent in term of number of iterations and the converged estimate between the two sets of data. The final errors in the diffusion coefficient estimates are $\varepsilon_1 = 4.8 \times 10^{-3}\%$ and $\varepsilon_1 = 1.04\%$, for clean and contaminated, respectively. The errors calculated based on the solution at the control point are $\varepsilon_2 = 6.6 \times 10^{-4}\%$ and $\varepsilon_2 = 0.16\%$. For demonstration purpose we show in Figure 5 the converged solution of the problem where the clean data is used.

Table 6: Example 2: Iterations for $Q = 7$, $D_0 = 0.006$ and $\Delta t = 0.1$ at $t = 0.5$.

# Itr	0% noise		5% noise	
	P_1	P_2	P_1	P_2
1	0.006600	0.006600	0.006600	0.006600
2	0.018109	0.009794	0.015941	0.009786
3	0.001385	0.009985	0.002308	0.009986
4	0.024000	0.010001	0.001500	0.010005
5	0.001500		0.002761	0.010010
6	0.023400		0.002968	
7	0.022800		0.003158	
8	0.005556		0.003180	
9	0.000435			
10	0.006217			
15	0.009776			
16	0.009957			
$\hat{D}^q_{opt}$	0.010000	0.010000	0.009922291	0.010103806
$\vartheta(\hat{D})$	6.46×10^{-07}	7.43×10^{-07}	2.94×10^{-04}	1.15×10^{-04}

4 Conclusions

In this paper, we present an approach for estimating the diffusion coefficient based on field measurements. The proposed algorithm involves solving the forward problem using a best estimate for the coefficient and then improving this estimate by evaluating the difference between the forward solution and the measured data. Furthermore, the problem may require repeated solutions over many iterations in time to achieve the required time instant of the taken measurement. Thus, the problem can become highly computationally demanding when the measurements are only available at a late time. Hence, many timesteps are required in the solution of the forward problem. Furthermore, considering late instants can often leads to larger temporal errors affecting the solution which could delay the convergence of the inverse problem solution. Using the partition of unity method it is possible to significantly speed the solution process by employing coarse meshes with enriched basis functions for the forward problem. Since, the basis functions are time independent it is possible to reuse the finite element linear system of equations at all time steps by only updating the right hand side. Hence, the solution for many time steps also becomes a lot more efficient. After evaluating the difference in the forward solution and the measured data, the secant and descent methods are used to improve the estimate. Furthermore, a stopping criteria is proposed which is based on optimizing an objective function.

The proposed algorithm can still be utilized even when only one data point is available. However, in order to avoid contingency and ensure the uniqueness of the inverse problem solution, we suggest using a minimum of two data points which is the procedure followed in this work. A third point is also utilized as a control point to calculate the estimate error. The numerical results in this paper suggest that the data points can be located anywhere inside the domain close to centre or the exterior boundaries. Two numerical examples are studied, where an analytical solution is available only in one of them while a converged finite element solution is used to create the input data in the second example. The obtained results show that the proposed approach converges to an accurate estimate even when the measured data are contaminated with noise or the initial guess is an order of magnitude different than the actual diffusion coefficient. The results also suggest that the algorithm is sensitive to the number of timesteps required. If the data measurements are taken at a later time more iterations are needed before the algorithm converges to an accurate estimate.

It should be noted that the gradient descent methods, Tikhonov regularization or Bayesian inference techniques are commonly used techniques in the inverse problem solution. Using such regularization methods can be necessary when solving an inverse problem in order to force convergence if the numerical results of the inverse solver diverge. The numerical experiments in this paper shows that the

proposed method converges without using any regularization methods even when using data contaminated with noise, which suggests the stability of the proposed method.

Although we introduce the partition of unity enrichment technique for the solution of a first class inverse problem, the technique can also be useful for second class problems. Furthermore, the presented work on a constant diffusion coefficient identification, forms a solid foundation for future consideration of heterogeneous materials. As future work, we plan to extend the approach for recovering the diffusion coefficient in heterogeneous materials.

Acknowledgements

Our research is supported by National Natural Science Foundation of China, No.51775270, and Jiangsu Innovation Program for Graduate Education, No.KYLX16_0314. Financial support provided by the project of Qatar National Research Fund under the contract NPRP11S-1220- 170112 is gratefully acknowledged.

Conflict of interest statement

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

[1] Stefan Kaessmair and Paul Steinmann. Computational first-order homogenization in chemomechanics. *Archive of Applied Mechanics*, 88(1-2):271–286, 2018.

[2] GJ McRae, WR Goodin, and JH Seinfeld. Numerical solution of the atmospheric diffusion equation for chemically reacting flows. *Journal of Computational Physics*, 45(1):1–42, 1982.

[3] J Kostrowicki and HA Scheraga. Application of the diffusion equation method for global optimization to oligopeptides. *The Journal of Physical Chemistry*, 96(18):7442–7449, 1992.

[4] HW Jensen, SR Marschner, M Levoy, and P Hanrahan. A practical model for subsurface light transport. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 511–518. ACM, 2001.

[5] MV Klibanov. Two classes of inverse problems for partial differential equations. *Annals of the New York Academy of Sciences*, 661(1):93–111, 1992.

[6] M Tadi, AK Nandakumaran, and SS Sritharan. An inverse problem for Helmholtz equation. *Inverse Problems in Science and Engineering*, 19(6):839–854, 2011.

[7] Y Cenesiz and A Kurt. The solutions of time and space conformable fractional heat equations with conformable Fourier transform. *Acta Universitatis Sapientiae, Mathematica*, 7(2):130–140, 2015.

[8] S Zhu, P Satravaha, and X Lu. Solving linear diffusion equations with the dual reciprocity method in Laplace space. *Engineering Analysis with Boundary Elements*, 13(1):1–10, 1994.

[9] C Qu, S Zhang, and R Liu. Separation of variables and exact solutions to quasilinear diffusion equations with nonlinear source. *Physica D: Nonlinear Phenomena*, 144(1):97–123, 2000.

[10] Jinhui Jiang, Mohammed Seaid, M Shadi Mohamed, and Hongqiu Li. Inverse algorithm for real-time road roughness estimation for autonomous vehicles. *Archive of Applied Mechanics*, pages 1–16, 2020.

[11] A Kaushik, V Kumar, and AK Vashishth. An efficient mixed asymptotic-numerical scheme for singularly perturbed convection diffusion problems. *Applied Mathematics and Computation*, 218(17):8645–8658, 2012.

[12] A Cardone, R D'Ambrosio, and B Paternoster. Exponentially fitted IMEX methods for advection–diffusion problems. *Journal of Computational and Applied Mathematics*, 316:100–108, 2017.

[13] Y Yang, MF Knol, F van Keulen, and C Ayas. A semi-analytical thermal modelling approach for selective laser melting. *Additive Manufacturing*, 21:284–297, 2018.

[14] A Nouy, A Clement, F Schoefs, and N Moës. An extended stochastic finite element method for solving stochastic partial differential equations on random domains. *Computer Methods in Applied Mechanics and Engineering*, 197(51):4663–4682, 2008.

[15] JM Melenk and I Babuška. The partition of unity finite element method: basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):289–314, 1996.

[16] O Laghrouche and P Bettess. Short wave modelling using special finite elements. *Journal of Computational Acoustics*, 8(01):189–210, 2000.

[17] MS Mohamed, M Seaid, J Trevelyan, and O Laghrouche. A partition of unity FEM for time-dependent diffusion problems using multiple enrichment functions. *International Journal for Numerical Methods in Engineering*, 93:245–265, 2013.

[18] M Drolia, MS Mohamed, O Laghrouche, M Seaid, and J Trevelyan. Enriched finite elements for initial-value problem of transverse electromagnetic waves in time domain. *Computers & Structures*, 182:354–367, 2017.

[19] J Jiang, MS Mohamed, M Seaid, and H Li. Identifying the wavenumber for the inverse Helmholtz problem using an enriched finite element formulation. *Computer Methods in Applied Mechanics and Engineering*, 340:615–629, 2018.

[20] FP Van der Meer, R Al-Khoury, and LJ Sluys. Time-dependent shape functions for modeling highly transient geothermal systems. *International Journal for Numerical Methods in Engineering*, 77(2):240–260, 2009.

[21] P OHara, CA Duarte, T Eason, and J Garzon. Efficient analysis of transient heat transfer problems exhibiting sharp thermal gradients. *Computational Mechanics*, pages 1–22, 2013.

[22] M Shadi Mohamed, Mohammed Seaid, and Abderrahman Bouhamidi. Iterative solvers for generalized finite element solution of boundary-value problems. *Numerical Linear Algebra with Applications*, 25(6):e2205, 2018.

[23] M Malek, N Izem, MS Mohamed, M Seaid, and O Laghrouche. A partition of unity finite element method for three-dimensional transient diffusion problems with sharp gradients. *Journal of Computational Physics*, 396:702–717, 2019.

[24] Mustapha Malek, Nouh Izem, Mohammed Seaid, M Shadi Mohamed, and Mohamed Wakrim. A partition of unity finite element method for nonlinear transient diffusion problems in heterogeneous materials. *Computational and Applied Mathematics*, 38(2):31, 2019.

[25] Mustapha Malek, Nouh Izem, M Shadi Mohamed, and Mohammed Seaid. A three-dimensional enriched finite element method for nonlinear transient heat transfer in functionally graded materials. *International Journal of Heat and Mass Transfer*, 155:119804, 2020.

[26] C Farhat, I Kalashnikova, and R Tezaur. A higher-order discontinuous enrichment method for the solution of high Péclet advection–diffusion problems on unstructured meshes. *International journal for numerical methods in engineering*, 81(5):604–636, 2010.

[27] I Kalashnikova, R Tezaur, and C Farhat. A discontinuous enrichment method for variable-coefficient advection–diffusion at high Péclet number. *International Journal for Numerical Methods in Engineering*, 87(1-5):309–335, 2011.

[28] R Borker, Ch Farhat, and R Tezaur. A discontinuous Galerkin method with Lagrange multipliers for spatially-dependent advection–diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 327:93–117, 2017.

[29] GR Richter. An inverse problem for the steady state diffusion equation. *SIAM Journal on Applied Mathematics*, 41(2):210–221, 1981.

[30] JR Cannon and P DuChateau. An inverse problem for a nonlinear diffusion equation. *SIAM Journal on Applied Mathematics*, 39(2):272–289, 1980.

[31] IV Amirkhanov, E Pavlušová, M Pavluš, TP Puzynina, IV Puzynin, and I Sarhadov. Numerical solution of an inverse diffusion problem for the moisture transfer coefficient in a porous material. *Materials and Structures*, 41(2):335–344, 2008.

[32] L Huiping, Z Guoqun, N Shanting, and L Yiguo. Inverse heat conduction analysis of quenching process using finite-element and optimization method. *Finite elements in Analysis and Design*, 42(12):1087–1096, 2006.

[33] AJ Davies, DB Christianson, LCW Dixon, R Roy, and P Van Der Zee. Reverse differentiation and the inverse diffusion problem. *Advances in Engineering Software*, 28(4):217–221, 1997.

[34] A Shidfar, R Pourgholi, and M Ebrahimi. A numerical method for solving of a nonlinear inverse diffusion problem. *Computers & Mathematics with Applications*, 52(6-7):1021–1030, 2006.

[35] GK Kocur, YE Harmanci, E Chatzi, H Steeb, and B Markert. Automated identification of the coefficient of restitution via bouncing ball measurement. *Archive of Applied Mechanics*, pages 1–14, 2020.

[36] H Kunze and DL Torre. An inverse problem for a 2-D system of steady-state reaction-diffusion equations on a perforated domain. In *AIP Conference Proceedings*, volume 1798, page 020089. AIP Publishing, 2017.

[37] B Jin and W Rundell. An inverse problem for a one-dimensional time-fractional diffusion problem. *Inverse Problems*, 28(7):075010, 2012.

[38] Y Wang, A Hamad, and M Tadi. Evaluation of a diffusion coefficient based on proper solution space. *Journal of Thermal Science and Engineering Applications*, 11(1):011017, 2019.

[39] L Sun, X Yan, and T Wei. Identification of time-dependent convection coefficient in a time-fractional diffusion equation. *Journal of Computational and Applied Mathematics*, 346:505–517, 2019.

[40] CJS Alves, Ma J Colaço, VMA Leitão, NFM Martins, HRB Orlande, and NC Roberty. Recovering the source term in a linear diffusion problem by the method of fundamental solutions. *Inverse Problems in Science and Engineering*, 16(8):1005–1021, 2008.

[41] H Wei, W Chen, H Sun, and X Li. A coupled method for inverse source problem of spatial fractional anomalous diffusion equations. *Inverse Problems in Science and Engineering; Formerly Inverse Problems in Engineering*, 18(7):945–956, 2010.

[42] ZQ Zhang and T Wei. Identifying an unknown source in time-fractional diffusion equation by a truncation method. *Applied Mathematics and Computation*, 219(11):5972–5983, 2013.

[43] NH Tuan and E Nane. Inverse source problem for time-fractional diffusion with discrete random noise. *Statistics & Probability Letters*, 120:126–134, 2017.

[44] DA Murio. Stable numerical solution of a fractional-diffusion inverse heat conduction problem. *Computers & Mathematics with Applications*, 53(10):1492–1501, 2007.

[45] W Rundell and Z Zhang. Recovering an unknown source in a fractional diffusion problem. *Journal of Computational Physics*, 368:299–314, 2018.

[46] Edoardo Patelli, Yves Govers, Matteo Broggi, Herbert Martins Gomes, Michael Link, and John E Mottershead. Sensitivity or bayesian model updating: a comparison of techniques using the dlr airmod test data. *Archive of Applied Mechanics*, 87(5):905–925, 2017.

[47] YM Mao, WD Zhang, H Ouyang, and JF Lin. Input force estimation accounting for modeling errors and noise in responses. *Archive of Applied Mechanics*, 85(7):909–919, 2015.

[48] Douglas L Miller, Nadine B Smith, Michael R Bailey, Gregory J Czarnota, Kullervo Hynynen, Inder Raj S Makin, and Bioeffects Committee of the American Institute of Ultrasound in Medicine. Overview of therapeutic ultrasound applications and safety considerations. *Journal of ultrasound in medicine*, 31(4):623–634, 2012.

[49] Kenji Ohmoto, Naoko Yoshioka, Yasuyuki Tomiyama, Norikuni Shibata, Tomoya Kawase, Koji Yoshida, Makoto Kuboki, and Shinichiro Yamamoto. Comparison of therapeutic effects between radiofrequency ablation and percutaneous microwave coagulation therapy for small hepatocellular carcinomas. *Journal of gastroenterology and hepatology*, 24(2):223–227, 2009.

[50] CT Kelley. *Iterative methods for optimization*. Society for Industrial and Applied Mathematics, 1999.

[51] M Iqbal, H Gimperlein, MS Mohamed, and O Laghrouche. An a posteriori error estimate for the generalized finite element method for transient heat diffusion problems. *International Journal for Numerical Methods in Engineering*, 110(12):1103–1118, 2017.