Applied Mathematics and Computation 409 (2021) 126402

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Efficient numerical solution of boundary identification problems: MFS with adaptive stochastic optimization

G.M.M. Reddy[a], P. Nanda[a], M. Vynnycky[b,*], J.A. Cuminato[c]

[a] Department of Mathematics, BITS-Pilani Hyderabad Campus, Telangana 500078, India
[b] Mathematics Applications Consortium for Science and Industry (MACSI), Department of Mathematics and Statistics, University of Limerick, Limerick V94 T9PX, Ireland
[c] Department of Applied Mathematics and Statistics, Institute of Mathematical and Computer Sciences, University of São Paulo at São Carlos, PO Box 668, 13560-970 São Carlos, São Paulo, Brazil

ARTICLE INFO

Article history:
Received 9 December 2020
Revised 24 March 2021
Accepted 17 May 2021

Keywords:
A posteriori error estimator
Fundamental solutions
Inverse boundary identification
Adaptive stochastic strategy

ABSTRACT

In this article, we study a novel computational technique for the efficient numerical solution of the inverse boundary identification problem with uncertain data in two dimensions. The method essentially relies on *a posteriori* error indicators consisting of the Tikhonov regularized solutions obtained by the method of fundamental solutions (MFS) and the given data for the problem in hand. For a desired accuracy, the *a posteriori* error estimator chooses the best possible combination of a complete set of fundamental solutions determined by the location of the sources that are arranged in a particular manner on a pseudo-boundary at each iteration. Also, since we are interested in a stable solution, an adaptive stochastic optimization strategy based on an error-balancing criterion is used, so as to avoid unstable regions where the stability contributions may be relatively large. These ideas are applied to two benchmark problems and are found to produce efficient and accurate results.

1. Introduction

In the context of heat transfer, the boundary identification problem is a special class of inverse geometric problem for determining the location and shape of the boundary of a conducting body by means of thermal measurements, performed on an accessible part of the boundary. In particular, such problems arise in steel blast furnaces, where it is desired to monitor the corroded thickness of the accreted refractory wall based on the measurement of temperature and heat flux on an accessible part of the boundary or at some internal positions [1–4]. Some other areas where heat-related boundary identification problems may arise are the following:

- glass fibre manufacture [5]: molten glass is pulled vertically upwards and cut into lengths as it cools;
- glass car windscreens [5]: the molten glass shape is produced so that, when it cools, it has the desired shape;
- manufacture of ceramic products [6]: these take on a different shape during the cooling process.

[*] Corresponding author.
E-mail address: michael.vynnycky@ul.ie (M. Vynnycky).

https://doi.org/10.1016/j.amc.2021.126402
0096-3003/

A popular approach for solving boundary identification problems in general has been the method of fundamental solutions (MFS), whose simplicity over the other mesh methods [4] made it a suitable choice for solving such problems; a recent review of the MFS can be found in [7]. Nonetheless, one crucial aspect of how to solve inverse problems efficiently using the MFS remains almost unexplored. An efficient numerical approximation depends on a reliable method to determine its quality and an efficient algorithm for the solution of the discrete problem. This can be achieved by obtaining a sharp *a posteriori* error estimator, which depends on the given data in the problem and the numerical solution. In particular, for inverse problems such as those involving boundary identification, which suffer from stability issues, it is a nontrivial task, since they are ill-posed in the sense of Hadamard [8], i.e., any small change in the input data can lead to a drastic change in the solution; hence, it is imperative to use regularization techniques for stabilizing the computations.

Besides, it is not clear how to develop an adaptive algorithm, once an *a posteriori* error estimator is available in the context of the MFS; this is primarily due to certain open issues associated with this method. For example, the accuracy obtained by the MFS relies on a suitable placement of the source points [9,10]. Moreover, the matrix resulting from the MFS system will be highly ill-conditioned [11]; note that this is because the inverse problem is ill-posed rather than, as may be the case in well-posed problems, because a larger number of boundary points have been used [13]. Regularization techniques, such as the Tikhonov regularization method, the damped singular value decomposition method, or the truncated singular value decomposition method, etc., usually give better control on levels of numerical accuracy to the original problem [12]. Further, a good choice of regularization parameter is vital for stable results; this can be chosen based upon different rules, such as the discrepancy principle [12], L-curve criterion [3], generalized cross-validation technique [22], and so on. The effect of the aforementioned methods for regularization in conjunction with different procedures for choosing a suitable value of the regularization parameter on the solution error and on the condition number of the MFS matrix has been studied in [13].

Against this background, the contributions of this article are twofold and are as follows.

- We derive an *a posteriori* error estimator for the boundary identification problem. From a mathematical point of view, we first use the stability results [14] to obtain an *a priori* bound on the error. Then, linear independence and denseness properties of the set of fundamental solutions [15,16] are used to obtain the *a posteriori* error estimate, which basically accounts for the discretization error. To take into account the contributions due to different levels of added noise, we need to choose an appropriate stability estimator, which then constitutes the total *a posteriori* estimator. There are a few articles available that deal with *a posteriori* error estimates for direct problems [17–19] in context of the MFS, but no such MFS results are available for boundary identification. The behaviour of the estimator is examined under the influence of random noise and a statistical analysis is performed for the effectivity indices in two spatial dimensions (2D).
- In order to identify the moving boundary efficiently, an attempt has been made to develop a stochastic algorithm in two dimensions. This, in turn, requires a suitable location of the source points and good choice of a regularization parameter at each iteration. The idea is to find a suitable pseudo-boundary at a certain distance from the heat conduction domain at each iteration and to adapt the source points on it. The algorithm is based on solving a linear system, and hence computationally cheap compared to the existing nonlinear optimization tools like MINPACK routine `lmdif` [20] and MATLAB optimization toolbox routine `lsqnonlin` [21]. The obtained *a posteriori* error estimates are used as raw ingredients to develop the algorithm. For a given tolerance and a specified percentage of noise level on the contaminated data, this algorithm generates:

  (a) relatively optimal location of source points outside the domain;
  (b) the optimal regularization parameter, without using any traditional regularization criteria, like discrepancy principle (DP) [12], generalized cross-validation (GCV) [3], or L-curve (LC) method [22];
  (c) the fewest possible source and collocation points;
  (d) the best possible reconstructed boundary, in view of (a)-(c).

  To obtain an accurate and stable solution, it is necessary to balance the discretization and stability contributions of the error separately, which is an important feature of this algorithm, and according to the knowledge of the authors, it does not seem to be available presently in existing routines for the boundary identification problem.

We will apply these ideas to two transient 2D benchmark examples from literature [22,23]; both are studied under the influence of noise. We can note that they are the only ones available in 2D, although others are available in one dimension (1D) [3,24–28]; these, however, are unsuitable for the present purposes, as they treat initial data reconstruction also.

The article is organized as follows. The mathematical formulation, along with a solution strategy for the boundary identification problem, is introduced in Section 2. In Section 3, we briefly discuss the motivation for acquiring *a posteriori* error estimates, whilst in Section 4 we obtain them. In Section 5, we develop and test the adaptive algorithm on two examples; numerical conclusions are drawn based on the quantitative analysis. In Section 6, we draw overall conclusions.

2. Mathematical formulation and solution strategy

We consider the heat conduction domain as $\Omega_T = \{(x,y,t) : 0 < x < 1,\ 0 < y < p(x,t),\ 0 < t \le T\}$, as shown in Fig. 1, where $p(x,t)$ determines the moving boundary. Let $\partial\Omega_T = \Gamma_0 \cup \Gamma_1 \cup \Gamma_2$ denote the boundary of Ω_T, where

$$\Gamma_0 = \{(x,y,0) : 0 \le x \le 1,\ 0 \le y \le p(x,0)\}$$

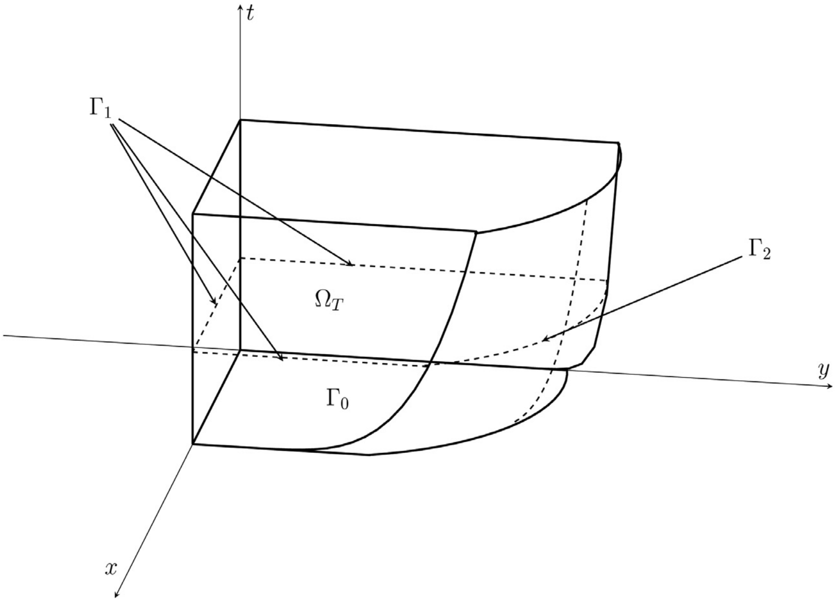


Fig. 1. Heat conduction domain Ω_T.

is the initial base,

$$\Gamma_1 = \{(x,0,t) : 0 \le x \le 1,\ 0 \le t \le T\} \cup \{(1,y,t) : 0 \le y \le p(1,t),\ 0 \le t \le T\} \cup \{(0,y,t) : 0 \le y \le p(0,t),\ 0 \le t \le T\}$$

is the part of the boundary on which Dirichlet and Neumann data are prescribed, and

$$\Gamma_2 = \{(x, p(x,t), t) : 0 < x < 1,\ 0 < t \le T\ \}$$

is the unknown moving boundary.

The canonical 2D heat equation is given as

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, \quad (x,y,t) \in \Omega_T \tag{1}$$

subject to the Dirichlet and Neumann data

$$u(x,y,t) = g_1^\delta(x,y,t), \quad (x,y,t) \in \Gamma_1 \tag{2}$$

$$\frac{\partial u}{\partial n}(x,y,t) = g_2^\delta(x,y,t), \quad (x,y,t) \in \Gamma_1, \tag{3}$$

where δ denotes the percentage of noise in the given data which might have been contaminated due to probable measurement error. Here, g_1^0 denotes the exact Dirichlet condition and g_2^0 denotes the exact Neumann condition. The boundary identification problem in this case involves locating the moving boundary $y = p(x,t)$, where

$$u(x,y,t) = u_p = 0, \quad (x,y,t) \in \Gamma_2.$$

Assuming purely Dirichlet conditions are prescribed on Γ_1, the following compatibility conditions must be satisfied: $g_1^0(0, p(0,t), t) = u_p$, $g_1^0(1, p(1,t), t) = u_p$, $0 \le t \le T$.

2.1. The MFS

We approximate the solution u as a linear combination of the fundamental solutions. The fundamental solution for the heat equation in 2D is given by

$$F(\mathbf{x}, t; \mathbf{z}, \tau) = \frac{H(t-\tau)}{4\pi(t-\tau)} \exp\left(-\frac{|\mathbf{x}-\mathbf{z}|^2}{4(t-\tau)}\right),$$

where $\mathbf{x}, \mathbf{z} \in \mathbb{R}^2$. Following the density arguments given in [15,16,29], the solution u of (1) can be approximated as follows:

$$u_{MFS}(\mathbf{x}, t) = \sum_{i=1}^{\mathcal{K}} \sum_{j=1}^{\mathcal{M}} c_j^i F(\mathbf{x}, t; \mathbf{z}^i, \tau_j), \quad (\mathbf{x}, t) \in \Omega_T,$$

where $\left(\mathbf{z}^i, \tau_j\right)$, $j = 1, \ldots, \mathcal{M}$, $i = 1, \ldots, \mathcal{K}$, are the source points, which are placed outside $\overline{\Omega}_T$, and c^i_j are the unknown coefficients to be determined.

2.1.1. Placement of source and collocation points

The source points are arranged on a pseudo-boundary which is at a positive distance h from the domain's boundaries. The shape of the pseudo-boundary is cylindrical and its cross-section at each time level $t \in [-T, T]$ is an up-scaled version of the initial base Γ_0. For a given $N \in \mathbb{N}$, let $M_1 = \lceil \frac{N}{2} \rceil$ and $M_2 = \lfloor \frac{N}{2} \rfloor$, where $\lceil\rceil$ and $\lfloor\rfloor$ denote the least integer and greatest integer functions, respectively. For placing the source points, we discretize the time-axis in $(-T, T)$ $\{0\}$ as

$$\tau_{M_1-m} = -\frac{T}{2N} - \frac{2T}{N+1}m, \quad m = 0, 1, \ldots, (M_1 - 1),$$

$$\tau_{M_1+n+1} = \frac{3T}{2N} + \frac{2T}{N}n, \quad n = 0, 1, \ldots, (M_2 - 1).$$

The spatial axes are discretized as

$$\alpha_{1_k} = -h + (k-1)\frac{1+2h}{N}, \quad k = 1, 2, \ldots, N,$$

$$\alpha_{2_k} = (1+h) - (k-1)\frac{1+2h}{N}, \quad k = 1, 2, \ldots, N,$$

and

$$\beta_{1_j} = -h + (j-1)\frac{p(1+h,0)+2h}{2N}, \quad j = 1, 2, \ldots, 2N,$$

$$\beta_{2_j} = (p(-h,0)+h) - (j-1)\frac{p(-h,0)+2h}{2N}, \quad j = 1, 2, \ldots, 2N.$$

For given N, the $6N^2$ source points are ordered as

$$\left(\alpha_{1_k}, -h, \tau_m\right), \; m = 1, 2, \ldots, N, \; k = 1, 2, \ldots, N,$$

$$\left(1+h, \beta_{1_j}, \tau_m\right), \; m = 1, 2, \ldots, N, \; j = 1, 2, \ldots, 2N,$$

$$\left(\alpha_{2_k}, p(\alpha_{2_k}, 0) + h, \tau_m\right), \; m = 1, 2, \ldots, N, \; k = 1, 2, \ldots, N,$$

and

$$\left(-h, \beta_{2_j}, \tau_m\right), \; m = 1, 2, \ldots, N, \; j = 1, 2, \ldots, 2N.$$

The boundary conditions (2) and (3) are collocated at $3N^2$ points on the fixed boundary Γ_1. For this purpose, the time axis is discretized as

$$t_i = \frac{i}{N+1}T, \; i = 1, 2, \ldots, N,$$

and the spatial axes are discretized as

$$x_k = \frac{k-1}{N}, \; k = 1, 2, \ldots, N,$$

$$y^i_{1_j} = \frac{j-1}{N}p(1, t_i), \; j = 1, 2, \ldots, N, \; i = 1, 2, \ldots, N,$$

$$y^i_{2_j} = \frac{N-j+1}{N}p(0, t_i), \; j = 1, 2, \ldots, N, \; i = 1, 2, \ldots, N.$$

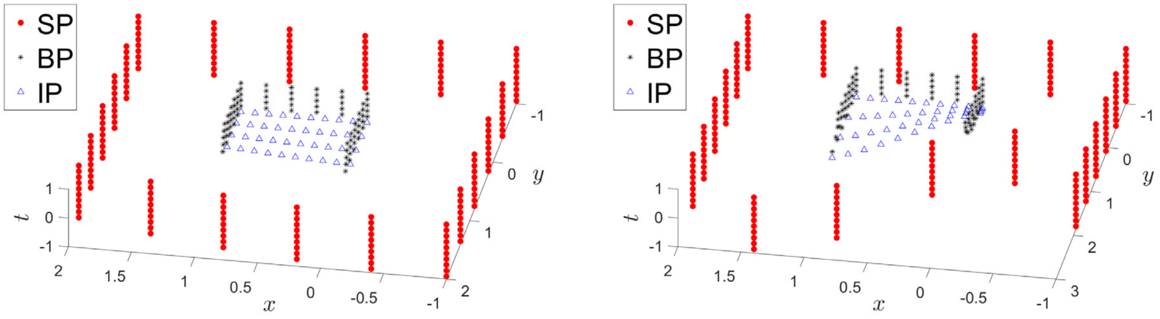
Moreover, the collocation points are arranged as

$$(x_k, 0, t_i), \; i = 1, 2, \ldots, N, \; k = 1, 2, \ldots, N,$$

$$\left(1, y^i_{1_j}, t_i\right), \; i = 1, 2, \ldots, N, \; j = 1, 2, \ldots, N,$$

$$\left(0, y^i_{2_j}, t_i\right), \; i = 1, 2, \ldots, N, \; j = 1, 2, \ldots, N.$$

Fig. 2a and b present the arrangement of source and collocation points for Examples 1 and 2, respectively, as described in Section 5.2.



(a) Example 1

(b) Example 2

Fig. 2. Source and collocation points. SP stands for the source points, whereas BP and IP stand for the collocation points on the lateral parts of the fixed boundary and on the initial base, respectively.

2.1.2. System of equations

We describe the system of equations as follows.

$$
\begin{aligned}
u_{MFS}(x_k, 0, t_i) &= g_1^{\delta}(x_k, 0, t_i), \; i = 1, \dots, N, \; k = 1, \dots, N,\\
u_{MFS}(1, y^i_{1_j}, t_i) &= g_1^{\delta}(1, y^i_{1_j}, t_i), \; i = 1, \dots, N, \; j = 1, \dots, N,\\
u_{MFS}(0, y^i_{2_j}, t_i) &= g_1^{\delta}(0, y^i_{1_j}, t_i), \; i = 1, \dots, N, \; j = 1, \dots, N,\\
-\frac{\partial u_{MFS}}{\partial y}(x_k, 0, t_i) &= g_2^{\delta}(x_k, 0, t_i), \; i = 1, \dots, N, \; k = 1, \dots, N,\\
\frac{\partial u_{MFS}}{\partial x}(1, y^i_{1_j}, t_i) &= g_2^{\delta}(1, y^i_{1_j}, t_i), \; i = 1, \dots, N, \; j = 1, \dots, N,\\
-\frac{\partial u_{MFS}}{\partial x}(0, y^i_{2_j}, t_i) &= g_2^{\delta}(0, y^i_{2_j}, t_i), \; i = 1, \dots, N, \; j = 1, \dots, N,
\end{aligned}
$$

The random noise has been added to the boundary conditions according to

$$g_k^{\delta}(x, t) = g_k^0(x, t) + \left(\delta \times \max_{(x,t)\in\Gamma_1} \left| g_k^0(x, t) \right| \times U(-1, 1) \right), \quad k = 1, 2,$$

where $U(-1, 1)$ denotes a random variable following the uniform distribution on $(-1, 1)$. The above-mentioned system of equations can be represented as

$$\mathbf{A}\mathbf{c} = \mathbf{f}, \tag{4}$$

where $\mathbf{c}$ denotes the vector of unknown coefficients c^i_j, $\mathbf{f}$ is the vector containing the collocated data, and $\mathbf{A}$ is the matrix containing the values of the fundamental solution evaluated for all possible combinations of the collocation and source points. The large condition number of the matrix $\mathbf{A}$ [11,30] underscores the importance of regularization. Let $\lambda > 0$ denote the regularization parameter and $\mathbf{I}$ the identity matrix. Applying the Tikhonov regularization technique results in the system

$$(\mathbf{A}^t\mathbf{A} + \lambda\mathbf{I})\mathbf{c} = \mathbf{A}^t\mathbf{f}, \tag{5}$$

where the superscript 't' denotes the transpose. Compared to (4), the system (5) is well-conditioned, and we use the Gaussian elimination method to solve it. Note that different regularization techniques such as the damped singular value decomposition, truncated singular value decomposition and Tikhonov regularization, in conjunction with the well-known schemes for choosing a suitable value of the regularization parameter namely the L-curve method, discrepancy principle, and the generalized cross validation, can also be employed. Here, although we apply Tikhonov regularization so as to regularize the ill-conditioned system (4), the proposed algorithm produces an optimal value of the regularization parameter λ without using any of the aforementioned well-known schemes.

3. Motivation for *a posteriori* error estimates

To construct a reliable and efficient method to determine the quality of approximation for the inverse boundary identification problem, we first need to establish a sharp *a posteriori* error estimate, and an efficient adaptive algorithm based on it. In order to construct a quality *a posteriori* error estimator it is necessary to identify and separate the error contributions. A further difficult question is how to use the obtained estimator to develop an efficient and adaptive algorithm. The approach adopted in this article is to find the optimal distance of the source points from the fixed boundary and to adapt the points on the pseudo-boundary at each iteration.

There are several methods available in the literature for finding the optimal distance of source points for different kinds of problems. One way to handle such an issue is the trial and error method. However, this approach is quite difficult as there are several parameters ($\mathcal{N}$, h, λ) for a given percentage of random noise. Recently, some approaches have been proposed using some optimizing tools. One such tool is MINPACK [20] routine `lmdif`. This routine implements a version of the *Levenberg-Marquardt* [31,32] algorithm and is designed to solve nonlinear least squares problems; however, it does not have the option to impose simple bounds on the handling variables. A part of the routine requires the computation of the Jacobian matrix for the objective function, the issues associated with which are discussed towards the end of this paragraph. Another tool is the MATLAB optimization toolbox routine `lsqnonlin` [15,21], which is designed to minimize the sum of squares of arbitrary differentiable functions. The routine uses two algorithms, namely the *trust-region-reflective* algorithm (which is effective for sparse problems) and the *Levenberg-Marquardt* algorithm, for this purpose. The former, which is the default setting for the routine, cannot handle underdetermined systems, and the latter, as mentioned earlier, cannot be applied when we have bound constraints. Furthermore, the *trust-region-reflective* algorithm, which is a gradient-based method, requires the computation of J^tJ, where J is the Jacobian matrix for the objective function. The user can either explicitly define the Jacobian or can input the sparsity pattern of the Jacobian matrix. However, that is possible, mostly, when the objective function is in a closed form. As an alternative, the algorithms use finite difference schemes in order to compute the Jacobian matrix. This could bring additional error into the system, while being computationally expensive.

One of the advantages of the above-mentioned approaches is that they only depend on the maximum possible *a posteriori* computable data available through the inverse problem. However, they suffer from some disadvantages:

- these algorithms are sensitive to initial inputs. If the initial guess is far from the exact solution, the computational time may be very large, and in some cases, it might not give good convergent results [15];
- with a good initial choice, they produce an optimal distance h, and hence a pseudo-boundary, but no information on the choice of the regularization parameter, λ, or on the number and location of source points, all of which play a significant role in determining the accuracy of the approximation [33].

This motivates us to establish some reliable and sharp *a posteriori* error indicators, which are capable of monitoring the desired error and useful in developing an adaptive algorithm that is:

- a transition from nonlinear to linear algorithms;
- suitable for more practical problems, where the input data must be independent of *a priori* quantities such as $p(x,t)$ and $u(x,y,t)$ [18].

In the next section, we shall derive *a posteriori* error estimates for the boundary identification problems.

4. *A posteriori* error estimates

The *a posteriori* error estimator to be developed in this section relies on the stability results of [14] along with density arguments of MFS approximations [16]. For $\omega_T \subset \overline{\Omega}_T$, we introduce the following notation:

$$\|\xi\|_{\infty,\omega_T} = \max_{(\mathbf{x},t)\in\omega_T} |\xi(\mathbf{x},t)|.$$

Further, let $u_j, j=1,2$ be two solutions of the heat equation (1) defined in $\{(x,y,t) : 0<x<1, 0<t<T, 0<y<p_j(x,t), j=1,2\}$. Suppose further that

$$u_j(x, p_j(x,t), t) = 0, \;\; 0<x<1, \;\; 0<t<T, \;\; j=1,2.$$

We begin with a two-dimensional version of the stability result derived in [14]. Interested readers may refer to [14] for the proof.

Lemma 1. *Assume:*

- *there exist two positive constants U_1 and U_2 such that*

$$\|u(\mathbf{x},t)\|_{\infty,\overline{\Omega}_T} \le U_1, \quad \text{and} \quad \min_{(\mathbf{x},t)\in\overline{\Omega}_T} u(\mathbf{x},t) \ge U_2 \;\; \forall\, t\in[0,T];$$

- *there exist two positive constants l and L such that*

$$l \le p(x,t) \le L;$$

- *for all (x,t) in the x-t plane and for all unit vectors $\mathbf{n}$ lying in the plane tangent to the surface p at (x,t), there exists a positive constant A such that*

$$\max\left|\nabla p(x,t)\cdot\mathbf{n}\right| < A.$$

Then, the moving boundary $p(x,t)$ depends continuously on the Dirichlet and Neumann data prescribed on Γ_1.

Now, to obtain an *a posteriori* error estimate based on this stability result, we need to exploit the approximation properties of fundamental solutions. In particular, we need to generate a linearly independent and dense set (total set) in $L^2(\Gamma_1)$. In this context, we shall briefly recall the following approximation properties of the set of fundamental solutions [16].

Lemma 2. *The set of fundamental solutions* $\{F(\mathbf{x},t;\mathbf{y_i},\tau_j)\}, i=1,2,\cdots; j=1,2,\cdots$ *forms a linearly independent and dense set in* $L^2(\Gamma_1)$*. Also, the set of normal derivatives* $\{\frac{\partial F}{\partial \mathbf{n}}(\mathbf{x},t;\mathbf{y_i},\tau_j)\}, i=1,2,\cdots; j=1,2,\cdots$ *forms a linearly independent and dense set in* $L^2(\Gamma_1)$.

***A posteriori* error estimate:** We now introduce two *a posteriori* error indicators, η^D and η^S.

The error indicator η^D represents the contribution mainly due to the discretization error, and is given by

$$\eta^D(u^{reg}_{MFS,\delta},g^\delta_1,g^\delta_2)=\left\|\sum_{i=1}^{\mathcal{K}}\sum_{j=1}^{\mathcal{M}}c^{i,reg}_jF\big(\mathbf{x},t;\mathbf{y_i}(\tau_j),\tau_j\big)-g^\delta_1(\mathbf{x},t)\right\|_{\infty,\Gamma_1}$$
$$+\left\|\sum_{i=1}^{\mathcal{K}}\sum_{j=1}^{\mathcal{M}}c^{i,reg}_j\frac{\partial F}{\partial \mathbf{n}}\big(\mathbf{x},t;\mathbf{y_i}(\tau_j),\tau_j\big)-g^\delta_2(\mathbf{x},t)\right\|_{\infty,\Gamma_1}, \tag{6}$$

where coefficients $c^{i,reg}_j$ result as the solution of the regularized system (5) and $\mathcal{M}$ is the number of source points placed on each $\mathbf{y_i}$.

The error indicator η^S reflects the stability error contributions due to added noise, and is given by

$$\eta^S(k)=\|p^{reg}_{MFS,k}-p^{reg}_{MFS,0}\|_{\infty,\Gamma_2}, \tag{7}$$

where $p^{reg}_{MFS,k}$ denotes the boundary identified using the regularized MFS with $k\%$ of added relative noise.

The *a posteriori* result can be stated as follows.

Theorem 1. *Under the assumptions of Lemma 1, we obtain an a posteriori error estimate of the form*

$$\eta=\eta^D+\eta^S$$

for the boundary identification problem, where the estimators η^D *and* η^S *are given by (6) and (7), respectively.*

Proof. Lemma 1 asserts that, quantitatively, $\|p_1(x,t)-p_2(x,t)\|_{\infty,\Gamma_2}$ depends continuously upon $\|u_1(\mathbf{x},t)-u_2(\mathbf{x},t)\|_{\infty,\Gamma_1}$ and $\|\frac{\partial u_1}{\partial \mathbf{n}}(\mathbf{x},t)-\frac{\partial u_2}{\partial \mathbf{n}}(\mathbf{x},t)\|_{\infty,\Gamma_1}$, where p_i, u_i, $i=1,2$, are as defined at the beginning of Section 4. If we assume that $p_1(x,t)$ corresponds to exact solution $p(x,t)$, and $p_2(x,t)$ is its MFS approximation $p_{MFS}(x,t)$, which has to deal with contaminated data, then the quantity of interest $\|p(x,t)-p_{MFS}(x,t)\|$ depends continuously upon $\|u(\mathbf{x},t)-g^\delta_1(\mathbf{x},t)\|_{\infty,\Gamma_1}$ and $\|\frac{\partial u}{\partial \mathbf{n}}(\mathbf{x},t)-g^\delta_2(\mathbf{x},t)\|_{\infty,\Gamma_1}$. Thus, the *a priori* estimate $\|u(\mathbf{x},t)-g^\delta_1(\mathbf{x},t)\|_{\infty,\Gamma_1}+\|\frac{\partial u}{\partial \mathbf{n}}(\mathbf{x},t)-g^\delta_2(\mathbf{x},t)\|_{\infty,\Gamma_1}$ gives a reliable bound for the error in identifying $p(x,t)$. In view of Lemma 2, this leads to an *a posteriori* error estimator with contaminated data $g^\delta_1(\mathbf{x},t)$ and $g^\delta_2(\mathbf{x},t)$ of the form

$$\left\|\sum_{i=1}^{\mathcal{K}}\sum_{j=1}^{\mathcal{M}}c^i_jF\big(\mathbf{x},t;\mathbf{y_i}(\tau_j),\tau_j\big)-g^\delta_1(\mathbf{x},t)\right\|_{\infty,\Gamma_1}+\left\|\sum_{i=1}^{\mathcal{K}}\sum_{j=1}^{\mathcal{M}}c^i_j\frac{\partial F}{\partial \mathbf{n}}\big(\mathbf{x},t;\mathbf{y_i}(\tau_j),\tau_j\big)-g^\delta_2(\mathbf{x},t)\right\|_{\infty,\Gamma_1}.$$

However, due to instability caused by the inverse problem, the regularization needs to be considered, which thus leads to η^D.

In addition, increasing the noise level is deleterious for the approximations, which can be controlled by the stability indicator η^S defined in (7), as shown in the next remark. □

Remark 1. We observe that the estimator η^D tends to get smaller with an increment in the basis functions, hence correctly reflecting the error $\|p(x,t)-p^{reg}_{MFS,0}\|_{\infty,\Gamma_2}$, which is expected to be small with a sufficiently large number of basis functions. However, under the influence of noise, say $k\%, k>0$, it is difficult to control the term $\|p(x,t)-p^{reg}_{MFS,k}\|_{\infty,\Gamma_2}$. The following observation has been made to control this term:

$$\begin{aligned}\|p(x,t)-p^{reg}_{MFS,k}\|_{\infty,\Gamma_2}&=\|p(x,t)-p^{reg}_{MFS,0}+p^{reg}_{MFS,0}-p^{reg}_{MFS,k}\|_{\infty,\Gamma_2}\\&\le\|p(x,t)-p^{reg}_{MFS,0}\|_{\infty,\Gamma_2}+\|p^{reg}_{MFS,k}-p^{reg}_{MFS,0}\|_{\infty,\Gamma_2}\\&\le\eta^D+\eta^S.\end{aligned}$$

It reflects the contribution of the error coming from stability component η^S.

5. Adaptive algorithm and numerical investigations

In this section, we study an adaptive algorithm for the efficient numerical solution of the boundary identification problems.

5.1. Adaptive algorithm

Algorithm 1 *A posteriori* stochastic algorithm for inverse boundary identification in 2D

```
Input: $\mathcal{N}_0, \mathcal{N}_{step}, h_{min}, h_{step}, h_{max}, \lambda_{min}, \lambda_{step}, \lambda_{max}, \delta, \texttt{TOL}$
Output: $\mathcal{N}_{opt}, h_{opt}, \lambda_{opt}, p^{reg}_{MFS,\delta}(\mathcal{N}_{opt}, h_{opt}, \lambda_{opt})$
1: $\mathcal{N} \leftarrow \mathcal{N}_0$
2: for $h \leftarrow h_{min} : h_{step} : h_{max}$ do                          ▷ $h_{step} = (h_{max} - h_{min})/\nu$
3:     for $\lambda \leftarrow \lambda_{min} : \lambda_{step} : \lambda_{max}$ do
4:         Compute $u^{reg}_{MFS,\delta}(\mathcal{N}, h, \lambda)$ when applied a random noise $\sim U(-1,1)$
                                                                              ▷ Solve
5:         Compute $\eta^D(\mathcal{N}, h, \lambda)$                          ▷ Estimate
6:     end for
7:     $\eta^D(\mathcal{N}, h, \lambda_h) = \min_{\lambda} \left( \eta^D(\mathcal{N}, h, \lambda) \right)$
8:     Compute $\eta^S(\mathcal{N}, h, \lambda_h)$
9: end for
10: Compute mean $\bar{X}$ and standard deviation $\bar{S}$ of the data set $\left\{ \frac{\eta^S(\mathcal{N},h,\lambda_h)}{\eta^D(\mathcal{N},h,\lambda_h)} \right\}_{h \in h_{min}:h_{step}:h_{max}}$.
11: Store $\mathcal{D} = \{\eta^D(\mathcal{N}, h, \lambda_h)\}_{h \in h_{min}:h_{step}:h_{max}}$
12: Compute $\eta^D(\mathcal{N}, h^*, \lambda_{h^*}) = \min \mathcal{D}$.
13: if $\frac{\eta^S(\mathcal{N},h^*,\lambda_{h^*})}{\eta^D(\mathcal{N},h^*,\lambda_{h^*})} < \bar{X} + \frac{t_{0.01,\nu-1}}{\sqrt{\nu}} \bar{S}$ then
14:     Go to Line 17
15: else $\mathcal{D} \leftarrow \mathcal{D} \setminus \{\eta^D(\mathcal{N}, h^*, \lambda_{h^*})\}$, Go to Line 12
16: end if
17: if $\eta^D(\mathcal{N}, h^*, \lambda_{h^*}) < \texttt{TOL}$ then
18:     $p^{reg}_{MFS,\delta}(\mathcal{N}_{opt}, h_{opt}, \lambda_{opt}) = p^{reg}_{MFS,\delta}(\mathcal{N}, h^*, \lambda_{h^*})$
19: else $\mathcal{N} \leftarrow \mathcal{N} + \mathcal{N}_{step}$, Go to Line 2
20: end if
```

Algorithm 1, presented in this segment, is based on the statistical information provided by choosing different input parameters. The input for the algorithm comprises:

- $\mathcal{N}_0$, the initial number of source points;
- $\mathcal{N}_{step}$, the increment for the number of source points;
- $[h_{min}, h_{max}]$, the range for the distance h of the pseudo-boundaries from the domain boundaries;
- h_{step}, the increment in h values;
- $[\lambda_{min}, \lambda_{max}]$, the range for the regularization parameter λ;
- λ_{step}, the increment in λ values;
- δ, the relative noise; and
- $\texttt{TOL}$, the tolerance on η^D.

With the above set of inputs, the aim of the algorithm is to produce the best possible estimated location of the moving boundary $p^{reg}_{MFS,\delta}$, the optimal distance of the source points from the boundaries h_{opt}, and the optimal regularization parameter λ_{opt} with the minimum possible computational effort.

As we start with a small number of source and collocation points, i.e. with a coarse approximation, we are likely to notice a high value of η^D and a smaller value of η^S. Here, the contribution towards error is more from the discretization than from the stability. With subsequent increase in the number of source points, the former is expected to decrease as a result of the addition of more basis functions in the MFS approximation. At the same time, due to the increment in size of the MFS matrix (**A** in (4)) and hence, the increment in its condition number, the stability component is expected to rise with some fluctuations. In order to have a stable solution, η^S should not be allowed to grow away from η^D and for an accurate solution, the latter should not be allowed to dominate the former by a large margin. Here, we try to study this delicate balance between the two estimators with the help of the quantity η^S/η^D.

To find an appropriate balance, we first minimize η^D, where we find the regions resulting in a small discretization error and among those regions, we select the one where the ratio η^S/η^D is not large. To elaborate, for a fixed number of source points $\mathcal{N}$, η^D is minimized over the λ-range for each of the h values in $[h_{min}, h_{max}]$. The λ values corresponding to these minimum values of η^D are marked and used to compute η^S for the given relative noise level. Statistical quantities such the mean, median and variance of η^S/η^D give good information on how to choose the range of h. Statistically, a range of the pseudo-boundary could be considered as reasonably good if, for the given range of h values, the mean and median show a good consistency and smaller variance of the particular quantity of interest. As it is of interest that discretization error must be in accordance with stability, so we shall be careful in choosing a good range of h, as there will be regions where the discretization errors may be small, but the stability contributions may be very high, and vice versa. For most of the computations, we consider the range of h as $[1.5, 2.3]$ with an increment of 0.1. The range of λ is kept as $[10^{-16}, 10^{-2}]$ with a multiplicative increment of 10.

If we were to adopt the deterministic approach, at this point, we would go for minimizing η^D again over h. Then, η^S could be handled with a different tolerance. However, we are interested in a bound on η^S/η^D which need not be specified explicitly. For the sake of simplicity, let us re-index the values of h in $[h_{min}, h_{max}]$, incremented by h_{step}, as $h_1, h_2, \ldots, h_\nu$, for some $\nu \in \mathbb{N}$. Further, let $X_1, X_2, \ldots, X_\nu$ be ν independent and identically distributed (i.i.d.) random samples where X_j is the random variable (RV) that takes on the value of η^S/η^D at $h_j,\ j = 1, 2, \ldots, \nu$.

Let $\bar{X} = \frac{1}{\nu}\sum_{i=1}^{\nu} X_i$ and $\bar{S}^2 = \frac{1}{\nu}\sum_{i=1}^{\nu}\left(X_j - \bar{X}\right)^2$ be the sample mean and the sample variance, respectively. It has been observed that the mean and median of the above random sample are sufficiently close to each other for the chosen range of h. Thus, it could be assumed that the i.i.d. samples come from a Normal population for a relatively large number of values of ν. Here, the sample mean could be seen as a probable bound for the desired η^S/η^D. As has been observed, the minimum η^D may sometimes correspond to a reasonable η^S, with their ratio being larger than the sample mean. These values of the estimators are likely to be leading to small error, but may get ignored if we just consider the sample mean as the bound. Similar cases may be encountered if we replace the sample mean by the estimated mean of the population. Therefore, a confidence upper bound of the population mean, μ, appears to be a more flexible and reliable choice.

More precisely, we consider the values of η^S and η^D for which their ratios fall within the 99% confidence upper bound for μ. Recorded are the ones corresponding to the minimum of those η^D. As a result, we would be ignoring the highly unstable solutions. As the number of simulated data points is small, i.e. 9, we assume a t-distribution for estimating the confidence bound. Moreover, the 99% confidence upper bound for μ would be $\left(\bar{X} + \frac{t_{0.01,8}}{3}\bar{S}\right)$. If we go with an upper bound with different confidence, say 95%, then that would be closer to the sample mean. Consequently, a sharper confidence bound may exclude some ratios where the error is likely to be good. Therefore, a 99% confidence bound appears to be a reasonable tolerance on η^S/η^D.

After choosing the error estimators with a suitable ratio, we could expect a stable solution, whereas the accuracy part yet remains to be examined. To this direction, we try imposing a tolerance $\texttt{TOL}$ on η^D, since it exhibits a monotonically decreasing behaviour with increment of basis functions compared to random behaviour of the stability estimator η^S. Upon satisfying the tolerance, the algorithm returns the parameters corresponding to the obtained estimator-values and computes the approximate moving boundary with the same. Otherwise, it moves on to the next iteration, thereby updating $\mathcal{N}$.

5.2. Numerical investigation

In this section, two benchmark problems are investigated. We impose $\delta = 1\%, 5\%, 10\%$ for the first example and $\delta = 1\%, 5\%$ for the second example, so as to study the effect of noise on the algorithm. We choose two nonlinear examples, one each from [22,23]. We choose the examples based on how the moving interface $p(x,t)$ is evolving with respect to time t, although both are nonlinear when considered with respect to x. In the first example, we choose a case where the moving surface $p(x,t)$ evolves linearly with respect to t with almost horizontal slope, whereas in the second example, $p(x,t)$ is a nonlinear function of time t, which exhibits a sharp behaviour around (0,0), with a greater slope with respect to t. This makes the second example much more difficult to handle, which is evident from the error profiles shown later in Fig. 15: from Fig. 15a and b, we observe that to obtain a particular desired accuracy for Example 2, we need a larger number of points, and hence more basis functions, compared to that for Example 1. Also, the Dirichlet and Neumann conditions on the faces are nonlinear for Example 2 and linear for Example 1.

For each of these examples and for each δ, the presented figures show the MFS approximation of the moving boundary p_{MFS} and the corresponding errors with different tolerances on η^D. Henceforth, the abbreviations *ABSe*, *RMSe*, and *RELe* stand for the maximum absolute error, the root mean squared error, and the relative error in the approximation, respectively.

Example 1. For this example, we consider the exact solution

$$u(x,y,t) = \frac{(x+1)^2}{20} + y + \frac{t}{10}, \quad (x,y,t) \in \Omega_T, \tag{8}$$

where $\Omega_T = \{(x,y,t) : 0 < x < 1,\ 0 < y < p(x,t),\ 0 < t \le 1\}$ is the heat conduction domain, and

$$p(x,t) = 1 - \frac{t}{10} - \frac{(x+1)^2}{20}, \quad 0 \le x \le 1,\ 0 \le t \le 1,$$

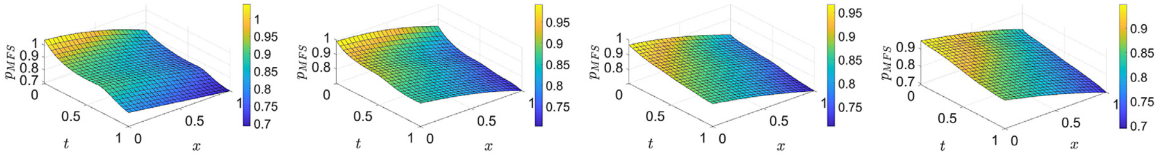
is the moving boundary. The Dirichlet and Neumann boundary conditions on the fixed boundaries and the initial condition at $t = 0$ can be obtained using (8).

Here, the temperature on the moving boundary is non-zero, i.e.,

$$u(x, p(x,t), t) = u_p = 1,\ 0 \le x \le 1,\ 0 \le t \le 1.$$

Example 2. The exact solution is given by

$$u(x,y,t) = e^{-4t}\sin(2x-1) - y + 1, \quad (x,y,t) \in \Omega_T,$$



(a) $TOL = 0.5$, $\mathcal{N}_{opt} = 54$, $h_{opt} = 2.0$, $\lambda_{opt} = 1.0e-06$

(b) $TOL = 0.3$, $\mathcal{N}_{opt} = 150$, $h_{opt} = 2.0$, $\lambda_{opt} = 1.0e-05$

(c) $TOL = 0.1$, $\mathcal{N}_{opt} = 294$, $h_{opt} = 2.3$, $\lambda_{opt} = 1.0e-07$

(d) $TOL = 0.06$, $\mathcal{N}_{opt} = 486$, $h_{opt} = 2.2$, $\lambda_{opt} = 1.0e-06$

Fig. 3. MFS approximation of p for different tolerances on η^D, Example 1, $\delta = 1\%$

where the domain of heat conduction taken as $\Omega_T = \{(x, y, t) : 0 < x < 1,\ 0 < y < p(x, t),\ 0 < t \le 1\}$.

Here, the moving boundary to be identified is given by

$$p(x, t) = e^{-4t} \sin(2x - 1) + 1, \quad 0 \le x \le 1,\ 0 \le t \le 1,$$

and

$$u(x, p(x, t), t) = u_p = 0,\ 0 \le x \le 1,\ 0 \le t \le 1,$$

is the temperature on the moving boundary $p(x, t)$.

5.3. Analysis of numerical experiments

In this section, we shall present the analysis obtained through the numerical results. First, we discuss the behaviour of the *a posteriori* error estimator, which can suggest how appropriate the estimator is for the boundary identification. We also discuss the sensitivity of the proposed approach with respect to the fictitious boundary; this is relegated to Appendix A.

Quality of the estimator: To begin with, we check the behaviour of the error estimates in Figs. 13 and 14, where η_{rms} denotes the total estimator with both of its components η^D and η^S calculated in the root mean squared sense. In particular for each $\mathcal{N}$, η^D has been minimized over all the (h, λ) pairs and η^S has been calculated with those h and λ causing the minimum η^D. Under the influence of noise, it is evident from Figs. 13 and 14 that both *RMSe* and η_{rms} are decreasing with increasing $\mathcal{N}$ (except at a few values of $\mathcal{N}$, where the stability estimator η^S is large). The placement of source points seems to be adapting appropriately at each iteration, as the error is decreasing beyond the amount of noise entered into calculations. The estimator also appears to be behaving in a manner similar to the error, even where the latter has sudden fluctuations. This is a preliminary study to understand the estimator and owing to the subsequent observations, as discussed in Section 5.1, we implement the stochastic Algorithm 1 for computing the estimators.

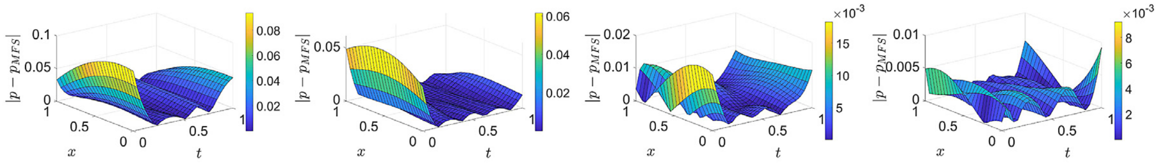
The quality of an *a posteriori* error estimator is often measured using a quantity called the effectivity index (EI). Due to the aforementioned additional issues, such as choosing an optimal regularization parameter and suitable placement of source points associated with the inverse boundary identification problem, at each iteration we define the effectivity index as

$$EI = \min_{h,\lambda} \frac{\eta}{\|p - p^{reg}_{MFS,\delta}\|},$$

where the percentage of δ is fixed, and all the quantities are measured in an RMS sense. At each iteration, the minimization is taken over η^D, the parameters $(h,\ \lambda)$ of which will then be used to calculate η^S and the error $\|p - p^{reg}_{MFS,\delta}\|$. Since we are dealing with noisy data, the deterministic values of EIs may not reflect the correct behaviour of an estimator. Therefore, we analyze the statistical behaviour of the EIs by fitting the obtained data. We fit the data with Generalized Extreme Value distribution (GEV) due to skewness of the data and the plots are shown in Figs. B1 and B2 in Appendix B. The maximum likelihood method is used to estimate the means. From Table B1, it is evident that η is the guaranteed upper bound on the error, as all statistical values of EIs are above 1. On the other hand, they are not far away from 1, showing the efficiency of the estimator. Alternatively, this is also evident in Figs. 13 and 14 as the gap between error and estimator is very narrow, though with a peak.

For Example 1, the boundary identification has been presented in Fig. 3a–d for 1% noise, in Fig. 5a–d for 5% noise, and in Fig. 7a–d for 10% noise. The error plots are provided in Fig. 4a–d for 1% noise, in Fig. 6a–d for 5% noise, and in Fig. 8a–d for 10% noise. Similarly for Example 2, we have shown the boundary identification in Fig. 9a–d for 1% noise, and in Fig. 11a–d for 5% noise with the errors being documented right below Fig. 10a–d, and Fig. 12a–d, respectively. The optimal parameters for each approximation have been listed below the corresponding boundary identification figure. By decreasing tolerances in Algorithm 1 as we are moving from left to right in Fig. 3a–d, the MFS approximations are becoming more accurate and the errors are decreasing, with the rightmost figure seeming to be very accurate, taking into account the error introduced into the computations due to noise. Similar observations can be made from the other figures.

The convergence plots for both of the examples are shown in Fig. 15. The solid curves represent the estimator η (computed in sup norm), whereas the dashed and the dotted lines represent the maximum absolute error (*ABSe*) and the root mean squared error (*RMSe*), respectively. We observe that for each percentage of noise (indicated by different colors), the



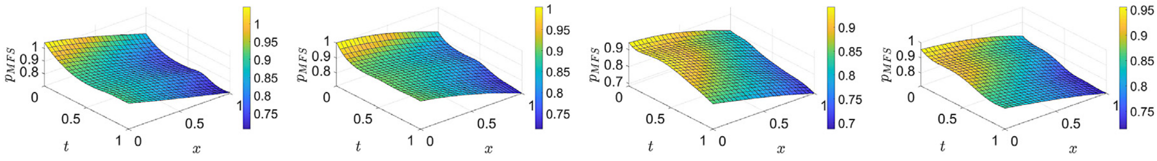
(a) $TOL = 0.5$, $ABSe = 0.093813$, $RMSe = 0.025653$, $RELe = 0.098750$

(b) $TOL = 0.3$, $ABSe = 0.061595$, $RMSe = 0.014514$, $RELe = 0.064837$

(c) $TOL = 0.1$, $ABSe = 0.018313$, $RMSe = 0.005419$, $RELe = 0.019277$

(d) $TOL = 0.06$, $ABSe = 0.009237$, $RMSe = 0.002321$, $RELe = 0.009723$

Fig. 4. Error in the MFS approximation for different tolerances on η^D, Example 1, $\delta = 1\%$



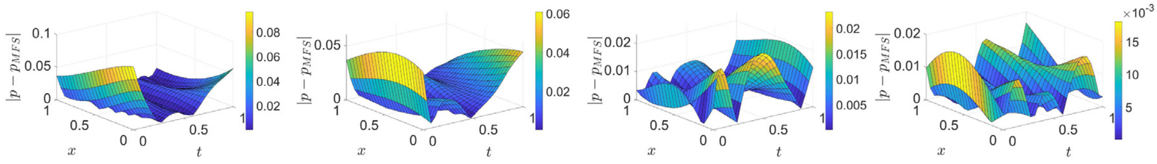
(a) $TOL = 0.6$, $\mathcal{N}_{opt} = 54$, $h_{opt} = 2.0$, $\lambda_{opt} = 1.0e-05$

(b) $TOL = 0.3$, $\mathcal{N}_{opt} = 150$, $h_{opt} = 2.0$, $\lambda_{opt} = 1.0e-04$

(c) $TOL = 0.1$, $\mathcal{N}_{opt} = 486$, $h_{opt} = 2.4$, $\lambda_{opt} = 1.0e-05$

(d) $TOL = 0.08$, $\mathcal{N}_{opt} = 1014$, $h_{opt} = 1.9$, $\lambda_{opt} = 1.0e-03$

Fig. 5. MFS approximation of p for different tolerances on η^D, Example 1, $\delta = 5\%$.



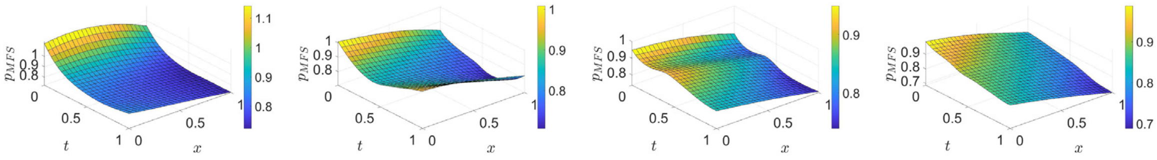
(a) $TOL = 0.6$, $ABSe = 0.096422$, $RMSe = 0.024731$, $RELe = 0.101497$

(b) $TOL = 0.3$, $ABSe = 0.060842$, $RMSe = 0.021189$, $RELe = 0.064044$

(c) $TOL = 0.1$, $ABSe = 0.023217$, $RMSe = 0.008863$, $RELe = 0.024439$

(d) $TOL = 0.08$, $ABSe = 0.018044$, $RMSe = 0.007814$, $RELe = 0.018994$

Fig. 6. Error in the MFS approximation for different tolerances on η^D, Example 1, $\delta = 5\%$.



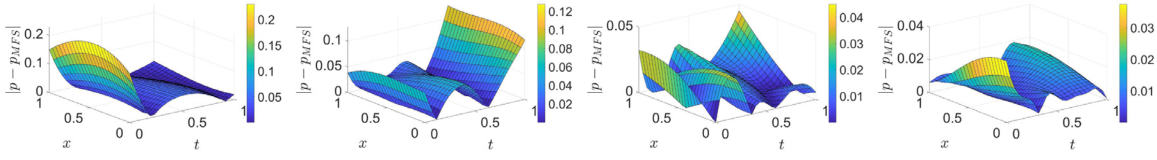
(a) $TOL = 0.6$, $\mathcal{N}_{opt} = 54$, $h_{opt} = 2.1$, $\lambda_{opt} = 1.0e-04$

(b) $TOL = 0.3$, $\mathcal{N}_{opt} = 150$, $h_{opt} = 1.8$, $\lambda_{opt} = 1.0e-04$

(c) $TOL = 0.2$, $\mathcal{N}_{opt} = 294$, $h_{opt} = 1.9$, $\lambda_{opt} = 1.0e-03$

(d) $TOL = 0.1$, $\mathcal{N}_{opt} = 1350$, $h_{opt} = 2.0$, $\lambda_{opt} = 1.0e-03$

Fig. 7. MFS approximation of p for different tolerances on η^D, Example 1, $\delta = 10\%$.



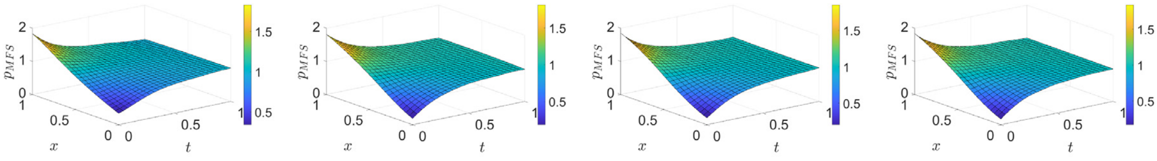
(a) $TOL = 0.6$, $ABSe = 0.229427$, $RMSe = 0.064833$, $RELe = 0.241502$

(b) $TOL = 0.3$, $ABSe = 0.127642$, $RMSe = 0.048539$, $RELe = 0.134360$

(c) $TOL = 0.2$, $ABSe = 0.044920$, $RMSe = 0.016084$, $RELe = 0.047285$

(d) $TOL = 0.1$, $ABSe = 0.037520$, $RMSe = 0.011736$, $RELe = 0.039495$

Fig. 8. Error in the MFS approximation for different tolerances on η^D, Example 1, $\delta = 10\%$.



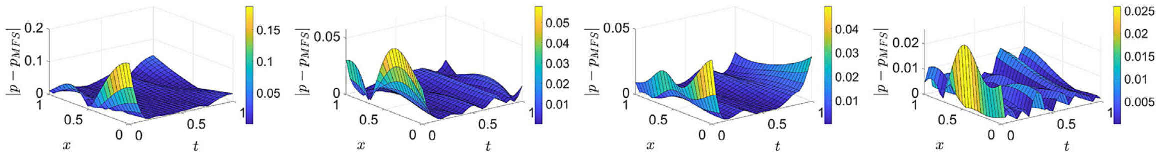
(a) $TOL = 0.6$, $\mathcal{N}_{opt} = 54$, $h_{opt} = 1.7$, $\lambda_{opt} = 1.0e-07$

(b) $TOL = 0.4$, $\mathcal{N}_{opt} = 150$, $h_{opt} = 2.0$, $\lambda_{opt} = 1.0e-06$

(c) $TOL = 0.25$, $\mathcal{N}_{opt} = 294$, $h_{opt} = 1.8$, $\lambda_{opt} = 1.0e-05$

(d) $TOL = 0.2$, $\mathcal{N}_{opt} = 486$, $h_{opt} = 2.1$, $\lambda_{opt} = 1.0e-07$

Fig. 9. MFS approximation of p for different tolerances on η^D, Example 2, $\delta = 1\%$



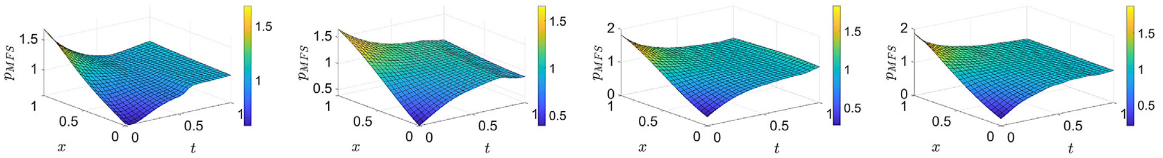
(a) $TOL = 0.6,\ ABSe = 0.186882,$ $RMSe = 0.024693,$ $RELe = 0.101485$

(b) $TOL = 0.4,\ ABSe = 0.057888,$ $RMSe = 0.011043,$ $RELe = 0.031436$

(c) $TOL = 0.25,\ ABSe = 0.049378,$ $RMSe = 0.008805,$ $RELe = 0.026814$

(d) $TOL = 0.2,\ ABSe = 0.025900,$ $RMSe = 0.005557,$ $RELe = 0.014065$

Fig. 10. Error in the MFS approximation for different tolerances on η^D, Example 2, $\delta = 1\%$



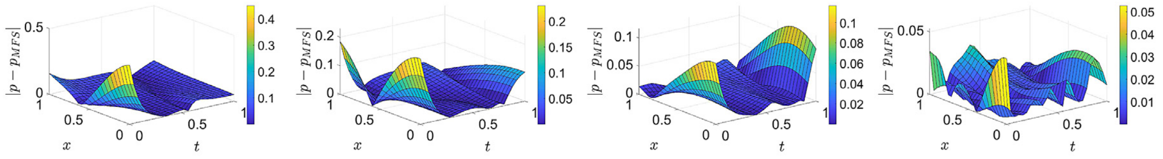
(a) $TOL = 0.8,\ \mathcal{N}_{opt} = 54,$ $h_{opt} = 1.8,\ \lambda_{opt} = 1.0e-05$

(b) $TOL = 0.6,\ \mathcal{N}_{opt} = 150,$ $h_{opt} = 2.2,\ \lambda_{opt} = 1.0e-05$

(c) $TOL = 0.4,\ \mathcal{N}_{opt} = 294,$ $h_{opt} = 1.9,\ \lambda_{opt} = 1.0e-04$

(d) $TOL = 0.25,\ \mathcal{N}_{opt} = 600,$ $h_{opt} = 1.5,\ \lambda_{opt} = 1.0e-03$

Fig. 11. MFS approximation of p for different tolerances on η^D, Example 2, $\delta = 5\%$



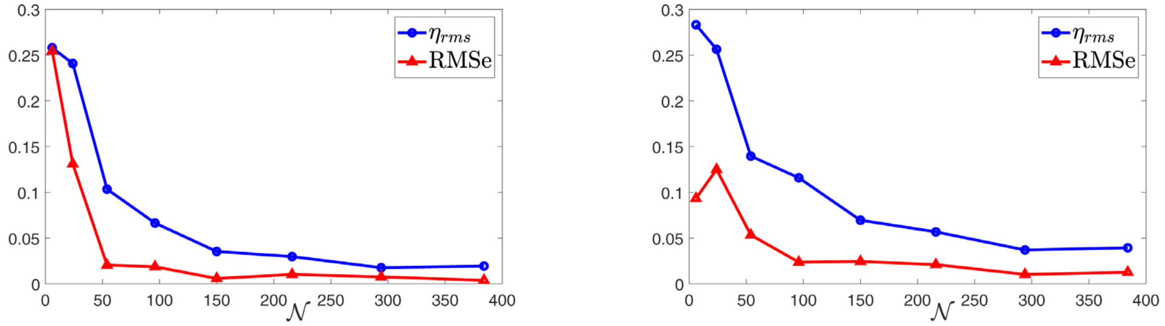
(a) $TOL = 0.8,\ ABSe = 0.450562,$ $RMSe = 0.072194,$ $RELe = 0.244675$

(b) $TOL = 0.6,\ ABSe = 0.230432,$ $RMSe = 0.049189,$ $RELe = 0.125135$

(c) $TOL = 0.4,\ ABSe = 0.116972,$ $RMSe = 0.030563,$ $RELe = 0.063521$

(d) $TOL = 0.25,\ ABSe = 0.052779,$ $RMSe = 0.014983,$ $RELe = 0.028661$

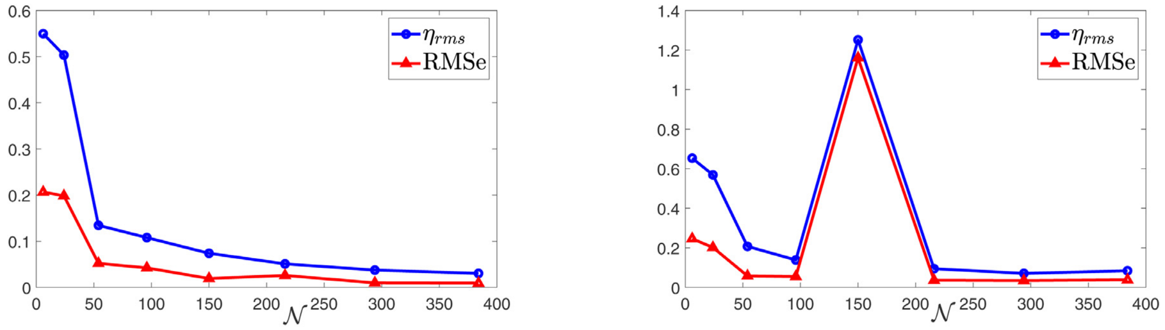
Fig. 12. Error in the MFS approximation for different tolerances on η^D, Example 2, $\delta = 5\%$



(a) $\delta = 1\%$

(b) $\delta = 5\%$

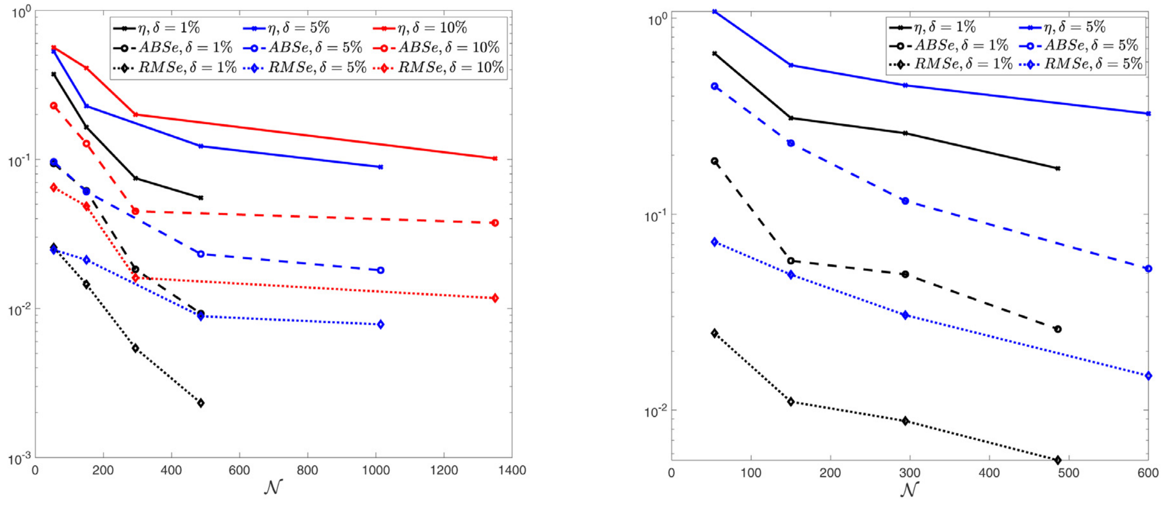
Fig. 13. Error and estimator behaviour for Example 1 using the deterministic approach.



(a) $\delta = 1\%$

(b) $\delta = 5\%$

Fig. 14. Error and estimator behaviour for Example 2 using the deterministic approach.



(a) Convergence for Example 1

(b) Convergence for Example 2

Fig. 15. Convergence with respect to different tolerances for different noise levels

estimator η is decreasing with increasing number of source points. A similar behaviour is depicted in both *ABSe* and *RMSe*. We can notice from Fig. 15a and b the change in behaviour of the error and estimator with respect to different percentages of noise. The curves for the estimator and errors with larger percentage of applied noise lie above those for smaller percentage of noise, as predicted for both examples. Further, for a given percentage of noise, the errors exhibit almost similar behaviour to the estimator. Such behaviour justifies that the proposed estimator can be a reliable bound for the error. Also, it is evident that a desired accuracy for a lower percentage of noise is attained with rather fewer points as compared with a high percentage of the noise, which shows the efficient behaviour of the estimator with respect to the noise level.

Remark 2. As we attempt to study the behaviour of discretization and stability errors separately, one possible way is to impose two different tolerances on them. Apparently, this approach would require the user to choose two different tolerances. For this purpose, a deterministic algorithm could be proposed as follows.

Algorithm 2 *A posteriori* deterministic algorithm for inverse boundary identification in 2D

```
Input: $\mathcal{N}_0, \mathcal{N}_{step}, h_{min}, h_{step}, h_{max}, \lambda_{min}, \lambda_{step}, \lambda_{max}, \delta, \mathtt{TOL}_D, \mathtt{TOL}_S$
Output: $\mathcal{N}_{opt}, h_{opt}, \lambda_{opt}, p^{reg}_{MFS,\delta}(\mathcal{N}_{opt}, h_{opt}, \lambda_{opt})$
 1: $\mathcal{N} \leftarrow \mathcal{N}_0$
 2: for $h \leftarrow h_{min} : h_{step} : h_{max}$ do                                   ▷ $h_{step} = (h_{max} - h_{min})/\#h$
 3:     for $\lambda \leftarrow \lambda_{min} : \lambda_{step} : \lambda_{max}$ do
 4:         Compute $u^{reg}_{MFS,\delta}(\mathcal{N}, h, \lambda)$ when applied a random noise $\sim U(-1, 1)$
                                                                                        ▷ Solve
 5:         Compute $\eta^D(\mathcal{N}, h, \lambda)$                                    ▷ Estimate
 6:     end for
 7:     $\eta^D(\mathcal{N}, h, \lambda_h) = \min_{\lambda}\left(\eta^D(\mathcal{N}, h, \lambda)\right)$
 8: end for
 9: $\eta^D(\mathcal{N}, h^{temp}, \lambda^{temp}) = \min_{h}\left(\eta^D(\mathcal{N}, h, \lambda_h)\right)$
10: if $\eta^D(\mathcal{N}, h^{temp}, \lambda^{temp}) < \mathtt{TOL}_D$ then
11:     Compute $p^{reg}_{MFS,\delta}(\mathcal{N}, h^{temp}, \lambda^{temp}), p^{reg}_{MFS,0}(\mathcal{N}, h^{temp}, \lambda^{temp}),\ \eta^S(\mathcal{N}, h^{temp}, \lambda^{temp})$
12:     if $\eta^S(\mathcal{N}, h^{temp}, \lambda^{temp}) < \mathtt{TOL}_S$ then
13:         $p^{reg}_{MFS,\delta}(\mathcal{N}_{opt}, h_{opt}, \lambda_{opt}) = p^{reg}_{MFS,\delta}(\mathcal{N}, h^{temp}, \lambda^{temp})$
14:     else $\mathcal{N} \leftarrow \mathcal{N} + \mathcal{N}_{step}$, Go to Line 2
15:     end if
16: else $\mathcal{N} \leftarrow \mathcal{N} + \mathcal{N}_{step}$, Go to Line 2
17: end if
```

The advantage of the stochastic algorithm (SA) over the deterministic algorithm (DA) can be perceived from Fig. 16. It exhibits the behaviour of optimal η, *ABSe* and *RMSe* produced by both deterministic and stochastic approaches for 6 values

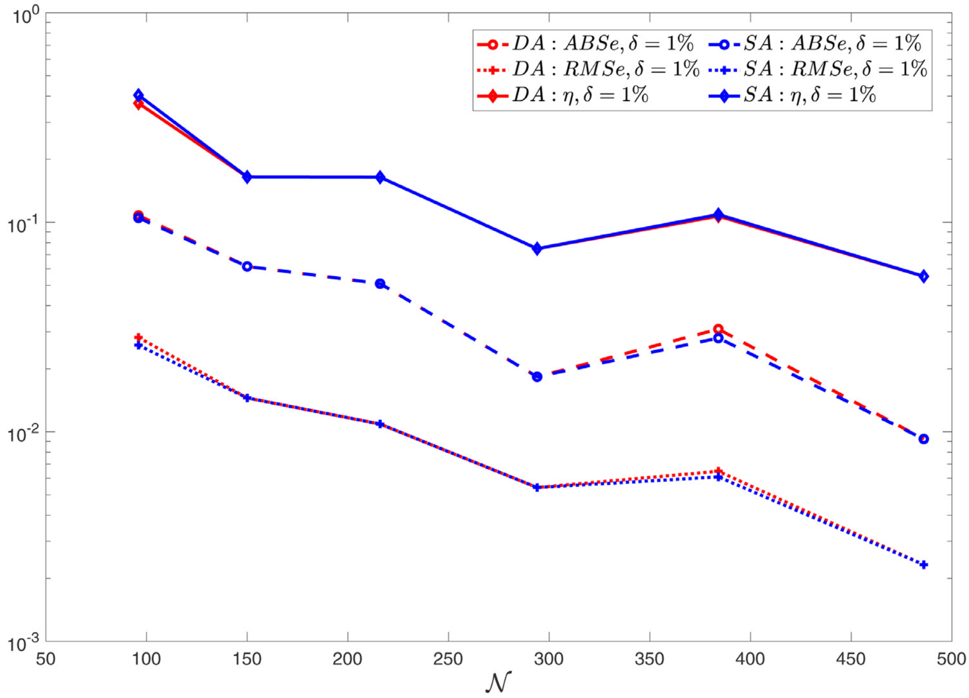


Fig. 16. Results for Example 1, $\delta = 1\%$ obtained with DA and SA.

of $\mathcal{N}$. With the deterministic approach, we recorded the results corresponding to the minimum of η^D (minimized over the $h - \lambda$ grid) without imposing any of the tolerances. Similarly for the stochastic approach, we did not put any tolerance on η^D, but we took that minimum η^D for which the respective η^S/η^D satisfies the confidence upper bound, as discussed in Section 5.1.

At the initial value of $\mathcal{N}$, η obtained by the deterministic approach (indicated by the solid red line) is below that obtained by the stochastic approach (indicated by the solid blue line). However, instead of recording the minimum η^D, the second method considers that value for which the proportion of η^S to η^D is balanced. Thus, possibly due to a bad η^S-value encountered by the deterministic approach, the ordering between the corresponding *ABSe* (indicated by the dashed lines) and *RMSe* (indicated by the dotted lines) is just the opposite to that between the respective estimators for the algorithms. A similar situation can be noticed at the fifth value of $\mathcal{N}$, though the gap between η-curves is very small there. Furthermore with the DA, it appears difficult to choose the right combination of the two tolerances. It has been observed from the computations that the minimum of η^D can also correspond to a value of η^S higher than those at other η^D values. This is probably due to the bad choice of the location of the basis functions and hence an increased condition number of the MFS matrix. Therefore, DA may ignore some good estimator values either due to violation of the tolerance on η^S or due to the minimization of η^D. On the other hand, the SA does not necessarily take the parameters corresponding to the minimum η^D and may give better results at an earlier iteration compared to DA, thus being more efficient. As an advantage, it comes with the reduced burden of choosing only one tolerance on η^D. In Fig. 16, the respective curves produced by deterministic and stochastic methods match at 4 values of $\mathcal{N}$. Thus, we expect the SA to perform better than or the same as the DA, if not worse.

Remark 3. The SA is expected to perform well in cases where the stability component of the estimator η^S takes on relatively smaller values than the discretization component η^D. As discussed in Section 5.1, the algorithm is based on the observation that the mean and median of the samples η^S/η^D are close to each other. Therefore, in cases where there is sudden and high fluctuation in η^S, the algorithm's performance may become affected. This oscillatory behaviour of η^S could depend upon the dimensions involved and the placement of source points that in turn decide the contributing basis functions in the MFS approximation.

5.4. Advantages and novelties of the method

Although MFS been used previously by others for boundary identification problems, this has normally been in association with a least squares formulation based on the discrepancy between the given and computed data; consequently, some objective functional can be minimized by employing appropriate nonlinear optimization algorithms. Here, on the other hand, we bypass these nonlinear routines by utilizing *a posteriori* estimators. To the best of our knowledge, there is no existing

Table 1
Comparison of the presented results with similar results from [22,23]. For Example 2, Liu and Wei [23] employed 10 points in the x-direction, around 50 points in the y-direction and 10 points in the t-direction, and hence around 5000 points in total.

	δ	Method	Number of points	*RMSe*
Example 1	5%	MFS [22]	900	0.0239
		a posteriori indicator	150	0.0212
Example 2	1%	Method of lines [23]	~ 5000	0.0597
		a posteriori indicator	54	0.0247
			150	0.011

Table 2
Time to complete one iteration of the stochastic algorithm for each of the test problems with $\delta = 1\%$.

	$\mathcal{N}$	Time (in seconds)
Example 1	54	137.96
	150	194.68
	294	388.69
	486	354.33
Example 2	54	141.10
	150	176.79
	294	604.39
	486	857.99

work which investigates the numerical solution of the boundary identification problem by implementing an algorithm based on *a posteriori* indicators and studies the efficiency of the MFS in terms of the minimum number of source points, the optimal distance of pseudo-boundary from the domain's boundaries, and the optimal regularization parameter.

Moreover, in Table 1, we provide a quantitative comparison of the benchmark results in [22,23] and those obtained using our method; from these, it is clear that our method supersedes the existing methods by some margin in terms of achieving lower error using fewer points.

5.5. Computational time

As regards the computing time for our algorithm, we note that it is difficult to make any general statement, as the time may vary depending on the MFS parameters used. However, in Table 2, we show the time required to complete one iteration of the stochastic algorithm for different values of $\mathcal{N}$. In particular, each iteration corresponds to a particular value of $\mathcal{N}$ and the computations are being done for the fixed ranges of values of h and λ, as taken earlier. Note that all computations were carried out on an Intel Core i7 Notebook with a 2.6 GHz 6-core processor and 16GB of RAM.

6. Conclusions and future work

In this article, we have considered the efficient numerical solution of the inverse boundary identification problem in 2D. We found that the estimator depicts robust behaviour with respect to error, and a statistical study of the effectivity index shows that the estimator is a reliable upper bound of the error. We demonstrated by numerical examples that the corresponding error indicator could be successfully used to construct a practical adaptive algorithm. It reconstructs the boundary efficiently by detecting the optimal distance of the source points from the boundary of the domain and optimal regularization parameter (without any traditional regularization approach such as the L-curve method, discrepancy principle or GCV, etc.). This might be thought of as an alternative to existing computationally expensive non-linear optimization tools like MINPACK or the MATLAB optimization toolbox routine `lsqnonlin`.

Since the purpose of the article was to build a novel methodology for simple and standard boundary identification problems, it will be interesting to see whether such an approach will succeed for other types of geometric inverse problems, such as cavity detection problems.

Acknowledgement

The authors would like to thank Prof. Sean McKee of the Department of Mathematics and Statistics, University of Strathclyde, for his insightful comments and suggestions. The first author would like to thank SERB, India for the financial support received [Grant Number SRG/2019/001973]. Further, the first and second authors acknowledge DST, New Delhi, India, for providing facilities through DST-FIST lab, Department of Mathematics, BITS-Pilani, Hyderabad Campus where a part of this work has been done.

Table A1
Error profiles and optimal parameter values with increased range of h, for Example 1, $\delta = 1\%$.

$\mathcal{N}_{opt}$	h_{opt}		λ_{opt}		*ABSe*		*RMSe*		*RELe*	
	Rng_1	Rng_2	Rng_1	Rng_2	Rng_1	Rng_2	Rng_1	Rng_2	Rng_1	Rng_2
54	2.0	2.0	10^{-6}	10^{-6}	0.093813	0.098468	0.025653	0.038664	0.098750	0.103650
150	2.0	2.3	10^{-5}	10^{-7}	0.061595	0.037214	0.014514	0.007408	0.064837	0.039173
294	2.3	1.7	10^{-7}	10^{-4}	0.018313	0.035399	0.005419	0.007964	0.019277	0.037262
486	2.2	2.2	10^{-6}	10^{-5}	0.009237	0.020015	0.002321	0.005643	0.009723	0.021069

Appendix A. Sensitivity of the fictitious boundary

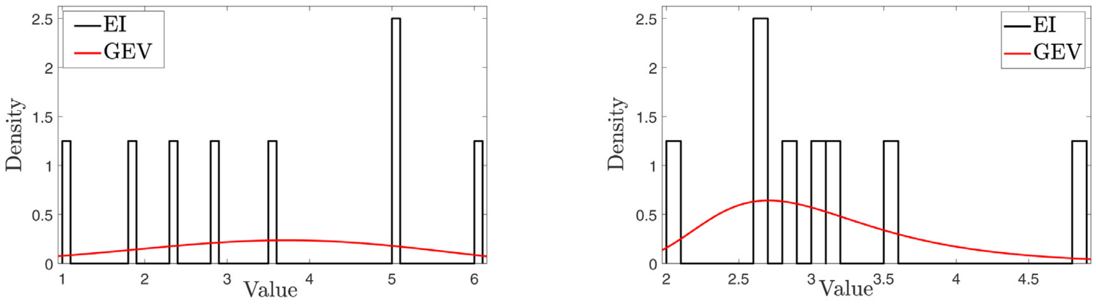
As is well-known [7], the MFS is sensitive to the fictitious boundary. In order to study this sensitivity, additional computations have been performed with an increased range of h-values, the distance of the pseudo-boundaries from the domain's boundaries. Precisely, for a given δ, we have run one iteration of the stochastic algorithm with $\mathcal{N} = \mathcal{N}_{opt}$, taking the the same range of λ as previously, and with the range of h changed to [1.5,3.0]. These computations have been conducted for Example 1 and $\delta = 1\%$. The optimal values of h and the corresponding error profiles for different values of $\mathcal{N}_{opt}$ are given in Table A1, wherein Rng_1 and Rng_2 represent the two chosen ranges of h-values, i.e., the one chosen for the earlier computations and [1.5,3.0], respectively.

Based on these results, we can make the following observations:

- Even after taking a larger range, the newly estimated optimal values of h fall within the interval [1.5,2.3] that has been chosen in most of the earlier computations. In some of those, the algorithm determined $h_{opt} = 2.3$. In such cases, we have translated the h-range, so as to have 2.3 in its interior, while keeping the number of h-values the same, i.e., 9.
- The reasons for restricting the number and range of h-values have been discussed in section 5.1, prior to Algorithm 1. Precisely, the allowed range is chosen so that the quantity of interest, η^S/η^D, can be treated as a Normal random variable.
- Allowing h to run over a larger set of numbers may have two potential shortcomings. First, it can be computationally expensive, as the algorithm will run over all the admissible λ-values for each of the additional h-values. This appears superfluous in view of the two items mentioned above. Second, with more h-values, the confidence upper bound employed in the computations may change. In particular, the term $\left(t_{0.01,\nu-1}\right)/\sqrt{\nu}$ in line 13 of Algorithm 1 may become smaller, thereby dragging the bound on η^S/η^D closer to the sample mean. This sharper bound is likely to retain the smaller values of η^S/η^D, which might result from a relatively larger value of η^D and a smaller value of η^S than those in the case of a relaxed upper bound as chosen in the earlier computations. Consequently, the accuracy may become compromised due to the higher values of η^D. This is evident from Table A1, where the error profiles corresponding to Rng_2 are larger than those corresponding to Rng_1, when $\mathcal{N}_{opt} = 294$ and 486.
- Another way of controlling the confidence upper bound on η^S/η^D is to change the confidence level, which is currently 99%. The possible effects of choosing a different upper bound for η^S/η^D have already been discussed in Section 5.1.
- It is observable from Table A1 that, with a larger set of h-values, we did not obtain improvement in the error profiles for $\mathcal{N}_{opt} = 54, 294$ and 486, although we got smaller error with the newly chosen h-range for $\mathcal{N}_{opt} = 150$.

In summary, it does not appear absolutely necessary to choose an increased number of permissible h-values, as it does not guarantee an improvement in the accuracy, but merely incurs a greater computational burden.

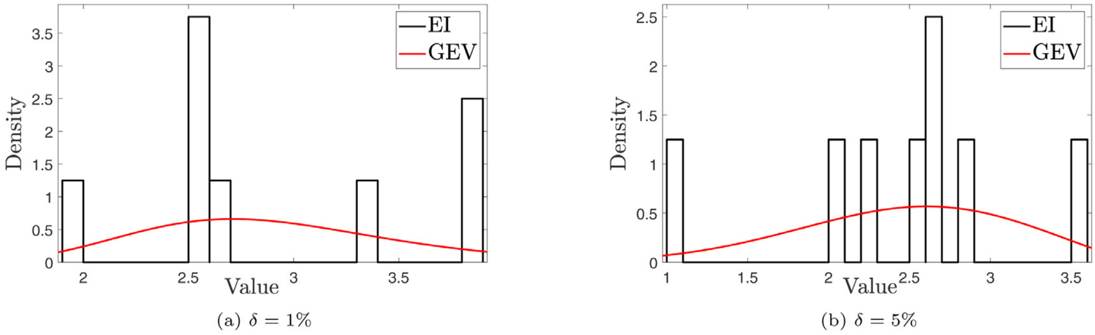
Appendix B. Statistical analysis of EIs



(a) $\delta = 1\%$

(b) $\delta = 5\%$

Fig. B1. Statistical behaviour of EI for Example 1.



(a) $\delta = 1\%$

(b) $\delta = 5\%$

Fig. B2. Statistical behaviour of EI for Example 2.

Table B1
Estimated parameter values for the distributions fitted to EI data. For the GEV distribution, μ denotes estimated location, σ denotes estimated scale parameter, and k denotes the estimated shape parameter.

Example	δ	μ	σ	k
1	0.01	2.96996	1.6931	-0.387144
	0.05	2.73644	0.57293	0.0578462
2	0.01	2.64525	0.559031	-0.110095
	0.05	2.25618	0.717524	-0.419859

References

[1] T.P. Fredman, A boundary identification method for an inverse heat conduction problem with an application in ironmaking, Heat Mass Trans. 41 (2) (2004) 95–103.
[2] N.L. Gol'dman, Inverse Stefan problems, Vol. 412 of Mathematics and its Applications, Kluwer Academic Publishers Group, Dordrecht, 1997.
[3] T. Wei, M. Yamamoto, Reconstruction of a moving boundary from Cauchy data in one-dimensional heat equation, Inverse Probl. Sci. Eng. 17 (2009) 551–567.
[4] A. Karageorghis, D. Lesnic, L. Marin, A survey of applications of the MFS to inverse problems, Inverse Probl. Sci. Eng. 19 (2011) 309–336.
[5] A. Farina, A. Klar, R.M.M. Mattheij, A. Mikelić, N. Siedow, Mathematical models in the manufacturing of glass, in: Mathematical Models in the Manufacturing of Glass, Vol. 2010 of Lecture Notes in Mathematics, Springer, 2011, pp. 1–227.
[6] T.K. Papathanasiou, F.D. Corso, A. Piccolroaz, Thermo-mechanical response FEM simulation of ceramic refractories undergoing severe temperature variations, J. Eur. Ceram. Soc. 36 (9) (2016) 2329–2340.
[7] A.H.D. Cheng, Y. Hong, An overview of the method of fundamental solutions-solvability, uniqueness, convergence, and stability, Eng. Anal. Bound. Elem. 120 (2020) 118–152.
[8] J. Hadamard, Lectures on Cauchy's problem in linear partial differential equations, Dover, 1952.
[9] C.J.S. Alves, On the choice of source points in the method of fundamental solutions, Eng. Anal. Bound. Elem. 33 (12) (2009) 1348–1361.
[10] C.S. Chen, A. Karageorghis, Y. Li, On choosing the location of the sources in the MFS, Numer. Algorithms 72 (1) (2016) 107–130.
[11] P.A. Ramachandran, Method of fundamental solutions: singular value decomposition analysis, Comm. Numer. Meth. Eng. 18 (11) (2002) 789–801.
[12] T. Wei, Y.C. Hon, L. Ling, Method of fundamental solutions with regularization techniques for Cauchy problems of elliptic operators, Eng. Anal. Bound. Elem. 31 (2007) 373–385.
[13] J. Lin, W. Chen, F. Wang, A new investigation into regularization techniques for the method of fundamental solutions, Math Comput. Simul. 81 (6) (2011) 1144–1152.
[14] P. Manselli, S. Vessella, On continuous dependence, on noncharacteristic Cauchy data, for level lines of solutions of the heat equation, Forum Math 3 (3) (1991) 513–522.
[15] T.H. Reeve, The method of fundamental solutions for some direct and inverse problems, Ph.D. thesis, University of Birmingham, UK, 2013.
[16] B.T. Johansson, Properties of a method of fundamental solutions for the parabolic heat equation, Appl. Math. Lett. 65 (2017) 83–89.
[17] C.J.S. Alves, P.R.S. Antunes, The method of fundamental solutions applied to boundary value problems on the surface of a sphere, Comp. Math Appl. 75 (7) (2018) 2365–2373.
[18] J. Oh, H. Zhu, Z. Fu, An adaptive method of fundamental solutions for solving the Laplace equation, Comp. Math Appl. 77 (7) (2019) 1828–1840.
[19] C.J.S. Alves, P.R.S. Antunes, N.F.M. Martins, S.S. Valtchev, Solving boundary value problems on manifolds with a plane waves method, Appl. Math. Lett. (2020) 106426.
[20] A. Karageorghis, D. Lesnic, Detection of cavities using the method of fundamental solutions, Inverse Probl. Sci. Eng. 17 (6) (2009) 803–820.
[21] A. Karageorghis, D. Lesnic, L. Marin, A moving pseudo-boundary method of fundamental solutions for void detection, Numer. Meth. Part. Diff. Eqs. 29 (3) (2013) 935–960.
[22] Y.C. Hon, M. Li, A computational method for inverse free boundary determination problem, Int. J. Numer. Meth. Eng. 73 (9) (2008) 1291–1309.
[23] J.C. Liu, T. Wei, Moving boundary identification for a two-dimensional inverse heat conduction problem, Inverse Probl. Sci. Eng. 19 (8) (2011) 1139–1154.
[24] Y.B. Wang, J. Cheng, J. Nakagawa, M. Yamamoto, A numerical method for solving the inverse heat conduction problem without initial value, Inverse Probl. Sci. Eng. 18 (2010) 655–671.
[25] C.-S. Liu, C.W. Chang, A simple algorithm for solving Cauchy problem of nonlinear heat equation without initial value, Int. J. Heat Mass Trans. 80 (2015) 562–569.
[26] J.-C. Liu, T. Wei, A quasi-reversibility regularization method for an inverse heat conduction problem without initial data, Appl. Math. Comp. 219 (2013) 10866–10881.
[27] C.S. Liu, An LGDAE method to solve nonlinear Cauchy problem without initial temperature, Comp. Mod. Eng. Sci. 99 (2014) 371–391.

[28] G.M.M. Reddy, P. Nanda, M. Vynnycky, J.A. Cuminato, An adaptive boundary algorithm to reconstruct initial and boundary data using the method of fundamental solutions for the inverse Cauchy-Stefan problem, Comp. Appl. Math. 40 (2021). 99 (26 pages).

[29] S. Chantasiriwan, B.T. Johansson, D. Lesnic, The method of fundamental solutions for free surface Stefan problems, Eng. Anal. Bound. Elem. 33 (4) (2009) 529–538.

[30] C.S. Chen, H.A. Cho, M.A. Golberg, Some comments on the ill-conditioning of the method of fundamental solutions, Eng. Anal. Bound. Elem. 30 (5) (2006) 405–410.

[31] K. Levenberg, A method for the solution of certain non-linear problems in least squares, Q. Appl. Math 2 (1944) 164–168.

[32] D.W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, J. Soc. Indust. Appl. Math. 11 (1963) 431–441.

[33] G.M.M. Reddy, M. Vynnycky, J.A. Cuminato, On efficient reconstruction of boundary data with optimal placement of the source points in the MFS: application to inverse Stefan problems, Inverse Probl. Sci. Eng. 26 (2018) 1249–1279.