



Multiphysics discovery with moving boundaries using Ensemble SINDy and peridynamic differential operator

Ali Can Bekar¹ · Ehsan Haghighat² · Erdogan Madenci¹

Received: 20 March 2024 / Accepted: 4 September 2024

© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2024

Abstract

This study proposes a novel framework for learning the underlying physics of phenomena with moving boundaries. The proposed approach combines Ensemble SINDy and Peridynamic Differential Operator (PDDO) and imposes an inductive bias assuming the moving boundary physics evolves in its own corotational coordinate system. The robustness of the approach is demonstrated by considering various levels of noise in the measured data using the 2D Fisher–Stefan model. The confidence intervals of recovered coefficients are listed, and the uncertainties of the moving boundary positions are depicted by obtaining the solutions with the recovered coefficients. Although the main focus of this study is the Fisher–Stefan model, the proposed approach is applicable to any type of moving boundary problem with a smooth moving boundary front without an intermediate zone of two states. The code and data for this framework is available at: https://github.com/alicanbekar/MB_PDDO-SINDy.

Keywords Model discovery · Peridynamic differential operator · SINDy · Moving boundary models · Ensemble learning

1 Introduction

Moving boundary problems are ubiquitous in engineering and biological systems such as melting or solidification [1, 2], tumor growth and wound healing [3], free surface flows [4], and electrophotography [5]. Usually, moving boundaries split the solution domain with different governing PDEs for each subdomain. Also, interface of the split regions obeys a different governing equation [5–7], often expressed in its own coordinate system. Therefore, the moving boundary of the domain is solved in addition to the field variables. These features make physical systems with moving boundaries an intriguing and challenging case for learning their governing PDEs from field measurements.

Stefan type problems represent a suitable model for learning in the context of moving boundaries. Stefan problem describes heat transfer and interface evolution between a liquid and a solid domain. They have long been used as a benchmark problem for numerical solvers with a rich literature [8]. Various implicit and explicit solvers were developed to solve forward and inverse Stefan problem [1, 9, 10]. Recently, Wang and Perdikaris [11] proposed a Physics-Informed Neural Network (PINN) solver for forward and inverse solutions of a Stefan type problem. They assume that either physics or data about the system are partially available, such as certain terms in the PDEs or sparse measurements. Hence, their approach is not applicable if the terms in the governing equation are not known. When the governing equation of the system is unknown and the only available information is the measured data, reliable and physically consistent predictions can be made by discovering the underlying physics of the system from the field measurements. Purely data driven methods such as Dynamic Mode Decomposition (DMD) also offers successful short-term predictions [12]. However, model discovery reveals interpretable models with more generalization capabilities [13].

Model discovery, particularly in the presence of noise is a challenging task. Bongard and Lipson [14] used a symbolic regression approach to discover governing equations of

✉ Erdogan Madenci
madenci@email.arizona.edu

Ali Can Bekar
acbekar@arizona.edu

Ehsan Haghighat
ehsanh@mit.edu

¹ The University of Arizona, Tucson, AZ 85721, USA

² Massachusetts Institute of Technology, Cambridge, MA 02139, USA

dynamical systems from measurements. Schaeffer [15] introduced the concept of sparse learning of PDEs. The key idea is to numerically calculate the time derivatives of the field data and create a matrix that comprises the spatial derivatives of the field data and subsequently cast the problem as a sparse regression between the two steps. Sparse regression is the mechanism to inflict parsimony. Schaeffer [15] solved sparse optimization with the Douglas-Rachford algorithm. This makes the framework robust to noise in time derivatives of the field variable. Brunton et al. [16] introduced Sparse Identification of Nonlinear Dynamics (SINDy). SINDy is a versatile framework with an efficient sparse optimization algorithm. They convert the nonlinear model identification to linear system of equations similar to [15] and solve for the sparse optimization by sequentially thresholding the least squares solutions to promote sparsity. In their influential work, Zhang and Schaeffer [17] analyzed the convergence behavior of SINDy, strengthening the theoretical foundation of the method. Subsequently, SINDy has been widely adopted in the field of learning PDEs and applied to a wide range of problems [18–21]. Inspired by [22], Messenger and Bortz extended the work to leverage the weak formulation of ODEs and PDEs (Weak SINDy) [23, 24]. This method eliminates the pointwise approximation of derivatives using the weak form integral and significantly increases noise tolerance of SINDy algorithm. Fasel et al. [25] combined ensemble learning with the SINDy algorithm and introduced Ensemble SINDy. They demonstrated the capabilities of Ensemble SINDy by recovering the coefficients of several PDEs from noisy and scarce measurements. While Ensemble SINDy may not be as effective as Weak SINDy in handling noise, it can be used for experiments with moving boundaries because of the pointwise and corotational nature of the the problem. Furthermore, Ensemble SINDy can be combined with Weak SINDY to enhance its performance.

Combining Ensemble SINDy and Peridynamic Differential Operator (PDDO) introduced by Madenci et al. [26], we propose a discovery framework for learning dynamics of moving boundaries. We assume the moving boundary physics is governed in its own corotational coordinate system, with normal and tangential directions to the boundary, and its derivatives are calculated using the PDDO. The PDDO enables differentiation through integration and does not require uniform sensor placement. It simply considers the interaction between neighboring points for the evaluation of derivatives, it can be used to calculate derivatives in any coordinate system by a straightforward modification and it integrates well with the existing PDE discovery methods [27]. There appears no study in the open literature addressing the discovery of governing equations of moving boundaries.

This study is organized as follows. In Sect. 2, we explain the Ensemble SINDy. In Sect. 3, we briefly describe the

multiphysics PDE model, Fisher–Stefan model and the numerical experiment to create training data synthetically. In Sect. 4, we present the results and performance of the proposed learning framework for different levels of noise in experimental data. Finally, we discuss the results and summarize the main conclusions in Sect. 5.

2 Ensemble SINDy

Introduced by Brunton et al. [16], SINDy algorithm for sparse learning of PDEs is based on

$$\arg \min_{\alpha} \|\mathbf{V} - \mathbf{F}\alpha\|_2^2 + \lambda \|\alpha\|_1, \quad (1)$$

in which \mathbf{F} and \mathbf{V} are the feature matrix and velocity vector, respectively. The vector α contains the unknown coefficients appearing in the PDE and λ is sparsity regularization parameter. An example of feature matrix \mathbf{F} , consisting of the field variable u and its spatial derivatives in one dimensional space, x can be constructed as

$$\mathbf{F} = \begin{bmatrix} 1 & u(x_1, t_1) & u_x(x_1, t_1) & \cdots & u_{xxx}^3(x_1, t_1) \\ 1 & u(x_2, t_1) & u_x(x_2, t_1) & \cdots & u_{xxx}^3(x_2, t_1) \\ 1 & u(x_3, t_1) & u_x(x_3, t_1) & \cdots & u_{xxx}^3(x_3, t_1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & u(x_{n-1}, t_m) & u_x(x_{n-1}, t_m) & \cdots & u_{xxx}^3(x_{n-1}, t_m) \\ 1 & u(x_n, t_m) & u_x(x_n, t_m) & \cdots & u_{xxx}^3(x_n, t_m) \end{bmatrix}, \quad (2)$$

where each column represents a different feature at n spatial points and m time instances (snapshots); the first column consists of unit values to accommodate for the bias term of the solution, i.e., a potential constant source. The velocity vector \mathbf{V} consisting of the time derivative of the field variable u is expressed as

$$\mathbf{V}^T = \{ u_t(x_1, t_1) \ u_t(x_2, t_1) \ u_t(x_3, t_1) \ \cdots \ u_t(x_{n-1}, t_m) \ u_t(x_n, t_m) \}. \quad (3)$$

The SINDy algorithm restates Eq. (1) as

$$\begin{aligned} \alpha^0 &= (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T \mathbf{V} \\ S^k &= \{ j \in [n_f] : |\alpha_j^k| \geq \lambda \}, \quad k \geq 0 \\ \alpha^{k+1} &= \arg \min_{\alpha \in \mathbb{R}^{n_f} : \text{sup}(\alpha) \subseteq S^k} \|\mathbf{V} - \mathbf{F}\alpha\|_2, \quad k \geq 0, \end{aligned} \quad (4)$$

where n_f is the number of features in the feature matrix. As shown by Zhang and Schaeffer [17], this algorithm presents attractive convergence features. However, it performs poorly in the presence of high correlation between the columns of feature library. On the other hand, STRidge algorithm is robust to the correlation between features. Therefore, Eq. (1) is modified to contain a penalty term, λ_2 and it can be recast as

$$\arg \min_{\alpha} \|\mathbf{V} - \mathbf{F}\alpha\|_2^2 + \lambda_1 \|\alpha\|_1 + \lambda_2 \|\alpha\|_2, \tag{5}$$

where λ_1 is the penalty parameter enforcing sparsity and λ_2 is the penalty parameter regularizing the magnitude of the recovered coefficients. The optimization of Eq. (5) can be achieved through the STRidge algorithm as

$$\begin{aligned} \alpha^0 &= (\mathbf{F}^T \mathbf{F} + \lambda_2 \mathbf{I})^{-1} \mathbf{F}^T \mathbf{V} \\ S^k &= \left\{ j \in [n_f] : |\alpha_j^k| \geq \lambda_1 \right\}, \quad k \geq 0 \\ \alpha^{k+1} &= \arg \min_{\alpha \in \mathbb{R}^{n_f} : \text{supp}(\alpha) \subseteq S^k} \|\mathbf{V} - \mathbf{F}\alpha\|_2 + \lambda_2 \|\alpha\|_2, \quad k \geq 0. \end{aligned} \tag{6}$$

This algorithm has been used successfully to recover governing nonlinear partial differential equations [28]. However, the success of this algorithm diminishes significantly in the presence of noise [28].

It is worth mentioning that the omission of higher-order terms in Eq. (2) is not due to any limitations of the method itself. While the method does not guarantee the discovery of the exact governing equation, it has been empirically proven effective in many cases. For instance, Shaeffer [15] notes that in the context of the KdV equation, phenomena such as traveling waves and solitons can create ambiguity, as the same dataset might be generated by different equations.

Fasel et al. [25] combined SINDy with ensemble learning to handle noisy measurements. Ensemble algorithms, which aggregate multiple hypotheses from a base learning algorithm, offer statistical, computational, and representational advantages over conventional algorithms such as the decision trees, k-nearest neighbors, support vector machines, and logistic regression [29]. Generally, the resulting ensemble algorithm is significantly more accurate than the original classifier. However, unstable learning algorithms i.e., learning algorithms which are sensitive to small changes in training dataset, should be used to leverage the full potential of ensemble methods.

Bagging (Bootstrap aggregating) [29] is one of the well known ensemble methods which is based on running the same learning method over different subsets of the same dataset. In Bagging, given a set of M data points, N distinct datasets are created by uniformly sampling M data points with replacement. The original algorithm is trained on generated datasets and finally, hypotheses are combined with aggregating the trained models by using the sample mean. Bragging (Bootstrap robust aggregating) uses the sample median for aggregation instead.

Fasel et al. [25] use bagging and bragging with Eq. (5) as the base learner. They demonstrate the capability of the proposed method by recovering the coefficients of several PDEs from noisy and scarce measurements. The instability of the learner is ensured by systematically removing features from the feature matrix and adding random noise to the dataset.

The main algorithm for the Ensemble SINDy approach is shown in Fig. 1.

In this study, Ensemble SINDy approach is applied to learning 2D Fisher–Stefan system. This system involves a reaction–diffusion equation along with a moving boundary equation referred to as the Stefan condition. First equation evolves in a Cartesian coordinate system while the Stefan condition is defined in reference to the coordinate system of the moving boundary. Therefore, feature matrices for the discovery of Stefan condition is constructed using derivatives with respect to the moving boundary corotational coordinate system.

3 2D Fisher–Stefan problem

This study concerns the recovery of the governing equation of the 2D Fisher–Stefan model [30]. Tam et al. [31, 32] recently conducted an extensive study on the 2D Fisher–Stefan model. This model has practical applications, such as representing cell tumor growth, fibroblast cells invading a partial wound or porous media modeling for population dynamics [33] making it tangible and important. We created the dataset using the open-source Julia code shared by Tam et al. [31, 32]. Also, the details of dataset generation is explained Appendix 3. 2D Fisher–Kolmogorov–Petrovsky–Piskunov (Fisher–KPP) equation is expressed as

$$\frac{\partial u}{\partial t} = \nabla \cdot (\nabla u) + u(1 - u), \text{ for } \mathbf{x} \in \Omega_1, \tag{7}$$

where $u(\mathbf{x}, t)$ denotes the population density extending the problem domain Ω_1 by spreading to the complement domain Ω_2 . As the moving boundary progresses towards Ω_2 , Ω_2 gradually shrinks, and Ω_1 expands accordingly. This expansion of Ω_1 represents the spreading of the population density into the complementary domain, effectively extending the problem domain.

It is assumed that the regions Ω_1 and Ω_2 are predefined and available at every timestep. In practical applications, these regions may require a priori segmentation from the observations. This segmentation can be done manually or by using a model like UNet, which is well-suited for such tasks.

This equation is a reaction–diffusion equation and has a logistic type of nonlinear source term on the right hand side $u(1 - u)$ which controls the growth and competition of the species u [34]. Inspired by the classical Stefan problem [8, 35], Fisher–Stefan problem is defined by coupling Fisher–KPP equation with a moving boundary $\Gamma(\mathbf{x}, t) = \partial\Omega_1 \cap \partial\Omega_2$. This boundary moves in its normal direction defined by the gradient of the population density. As a result, the problem domain Ω_1 evolves with the moving boundary. The Stefan and Dirichlet boundary conditions defined on the moving boundary are

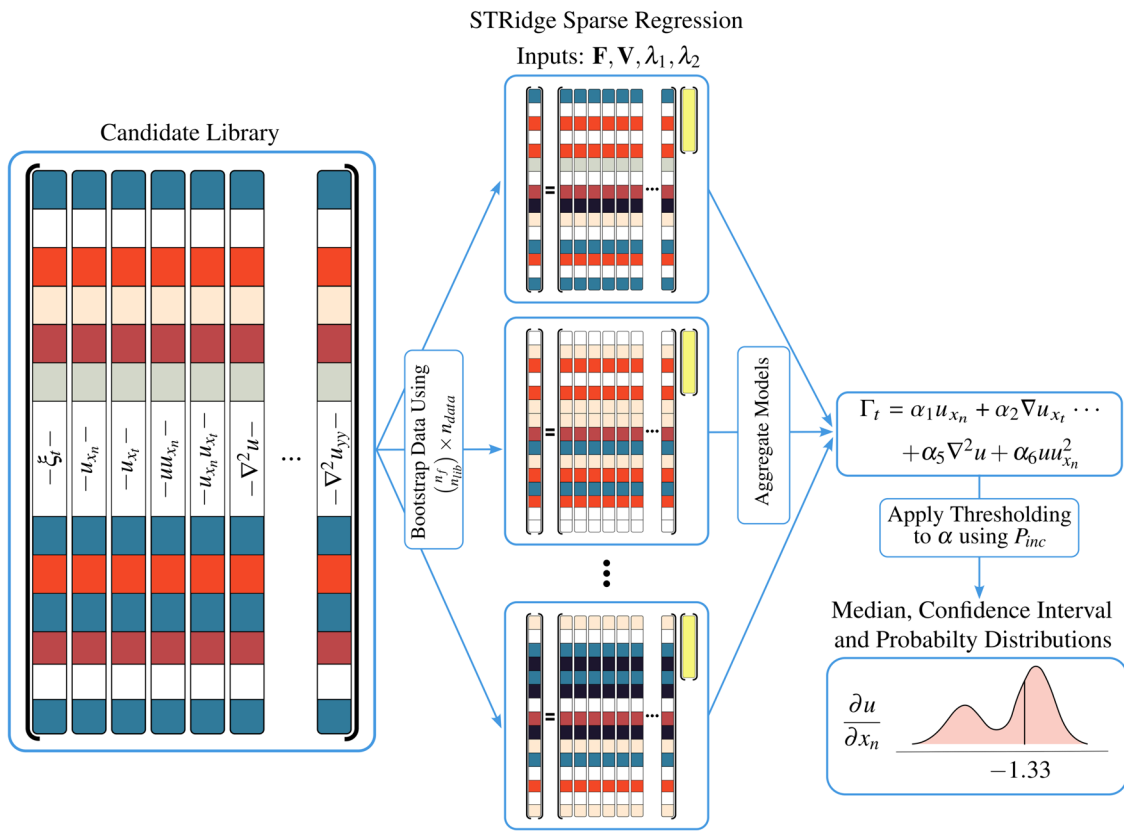


Fig. 1 Ensemble SINDy algorithm

$$\frac{\partial \Gamma_n}{\partial t} = -\kappa \nabla u \cdot \mathbf{n} \text{ and } u = u_f \text{ for } \mathbf{x} \in \Gamma, \tag{8}$$

where $\kappa > 0$ is the Stefan parameter, which connects the gradient of population density and surface evolution velocity. We consider two distinct test cases (datasets) for learning the governing equations of this system: (1) a vertical moving boundary with a sinusoidal perturbation, and (2) a circular moving boundary with irregularity.

3.1 Vertical moving boundary with a sinusoidal perturbation

Spatial domain for this problem is defined as $\mathbf{x} \in [0, 10] \times [0, 10][0, 10]$ and boundary conditions in y direction are defined as periodic. Initial-boundary conditions for this problem are depicted in Fig. 2.

The remaining boundary condition on the left edge representing a fixed population density and the periodic boundary conditions in y direction are expressed as

$$\begin{aligned} u(0, y, t) &= u_\ell \\ u(x, y + nL_y, t) &= u(x, y, t), \text{ for } n \in \mathbb{Z}, \end{aligned} \tag{9}$$

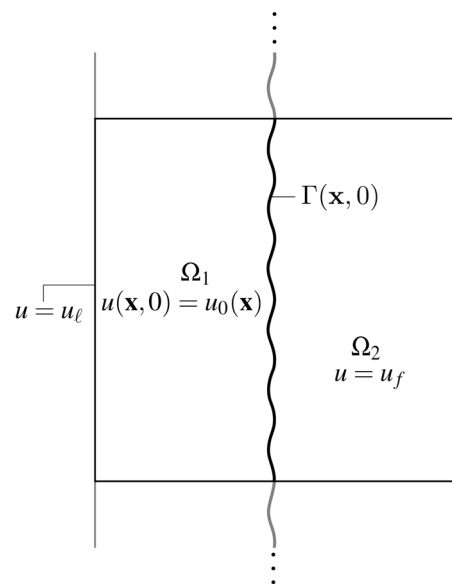


Fig. 2 Initial-boundary conditions for the 2D Fisher–Stefan problem with a vertical moving boundary having a sinusoidal perturbation

where $L_y = 10$ is the length of the domain in y direction. The initial conditions are defined as

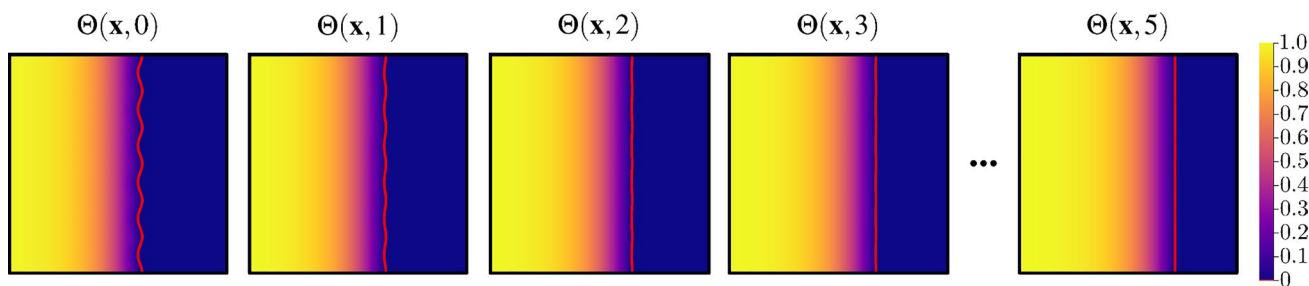


Fig. 3 Time evolution of the field data $\Theta = [u, \Gamma]$ for the 2D Fisher–Stefan problem with a vertical moving boundary having a sinusoidal perturbation

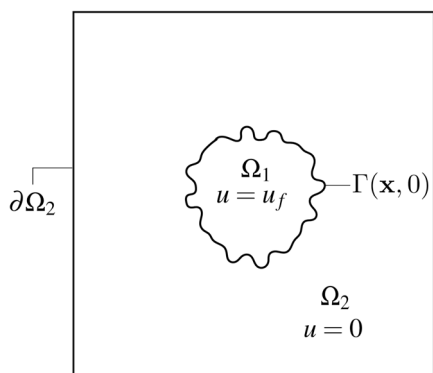


Fig. 4 Initial-boundary conditions for the 2D Fisher–Stefan problem with a circular moving boundary having irregularity

$$\begin{aligned}
 u(\mathbf{x}, 0) &= u_0(\mathbf{x}), \text{ for } \mathbf{x} \in \Omega_1 \\
 u(\mathbf{x}, 0) &= u_f, \text{ for } \mathbf{x} \in \Omega_2,
 \end{aligned}
 \tag{10}$$

where $u_0(\mathbf{x})$ is obtained by solving two overdetermined second order ordinary differential equations as described in [32]. The solution to the system of equations is obtained for $\kappa = 0.5, \Delta x = \Delta y = 0.025, \Delta t = 0.01, t \in [0, 5]$ using the open source code shared by [32] and the rest of the parameters of the problem can be also found in [32]. As time progresses, this solution for the population density along with the moving boundary is depicted in Fig. 3.

3.2 Circular moving boundary with irregularity

Figure 4 shows the spatial domain with $\mathbf{x} \in [-10, 10] \times [-10, 10]$ and the initial-boundary conditions.

The initial position of the moving boundary $\Gamma(\mathbf{x}, 0)$ is defined as

$$\Gamma(\mathbf{x}, 0) = 3.5 + 0.2(\sin 3\theta + \sin 12\theta + \sin 16\theta), \tag{11}$$

where

$$\theta = \tan^{-1} \frac{y}{x}. \tag{12}$$

The remaining boundary conditions on the outer edges representing fixed population density are expressed as

$$u(\mathbf{x}, 0) = 0 \text{ for } \mathbf{x} \in \partial\Omega_2. \tag{13}$$

The initial conditions are defined as

$$\begin{aligned}
 u(\mathbf{x}, 0) &= u_f, \text{ for } \mathbf{x} \in \Omega_1 \\
 u(\mathbf{x}, 0) &= 0, \text{ for } \mathbf{x} \in \Omega_2.
 \end{aligned}
 \tag{14}$$

The solution to the system of equations is obtained using the open source code shared by Tam and Simpson [31] for $\kappa = 0.1, \Delta x = \Delta y = 0.1, \Delta t = 0.04, t \in [0, 40]$. As time progresses, this solution for the population density along with the moving boundary is depicted in Fig. 5.

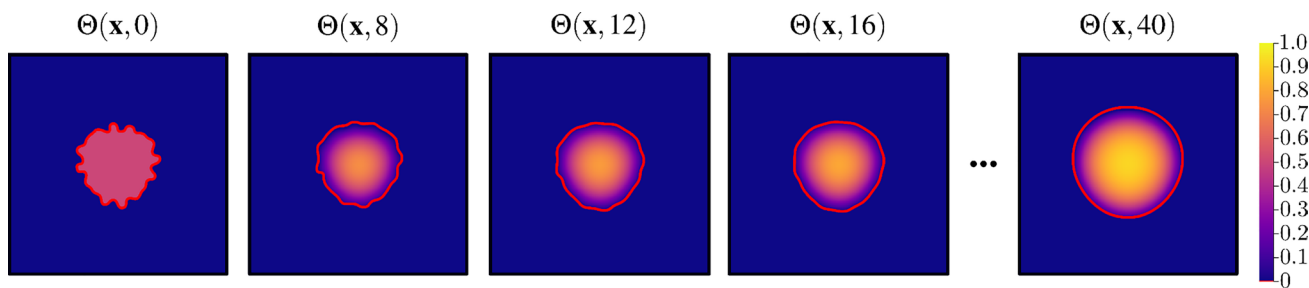


Fig. 5 Time evolution of the field data $\Theta = [u, \Gamma]$ for the 2D Fisher–Stefan problem with a circular moving boundary having irregularity

3.3 Corotational coordinate system of the moving boundary and the surface evolution velocity

This section describes the learning of the underlying governing equations for the 2D Fisher–Stefan problem. The learning includes the 2D Fisher–KPP equation which evolves in a 2D Cartesian coordinate system and the governing equation for the moving boundary (Stefan condition) which evolves in its own corotational coordinate system.

The derivatives of the field are constructed by employing the PDDO [26, 41] as explained in Appendix 1. For the derivatives in Cartesian coordinate system, the approach described by Bekar and Madenci [27] is adopted without any special treatment. The derivatives in corotational coordinates are calculated by rotating the coordinate system to the normal and tangent directions to the interface at the “sensor” locations. The “sensor” locations refer to imaginary nodes situated on the moving boundary as well as on the nodes of discretization within the domain to draw analogy to measurement locations in a real experimental setup. The families of the material points (representing the unique set of points interacting with the point of interest inside a finite radius) are also demarcated using the tangents. This approach is described in Fig. 6.

Coordinate transformation can be accomplished by using two different methods. The first method is based on the use of arctangents of the line sections between sensor locations on the moving boundary. The second method is based on the use of level sets. Using the snapshots of the moving boundary, the level set equation is solved and the gradients of level sets are calculated. These gradients after normalization define the normal directions to the moving boundary (interface).

Figure 7 illustrates the coordinate system rotation and advancing interface velocity calculations using panels

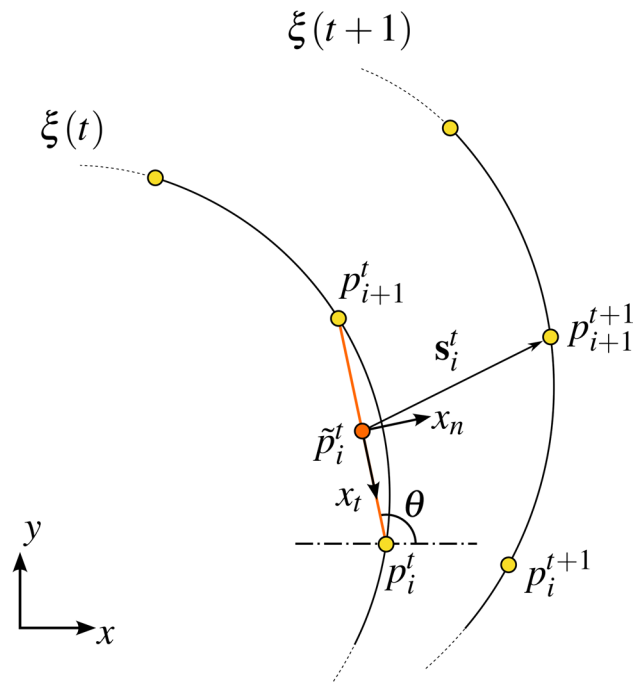
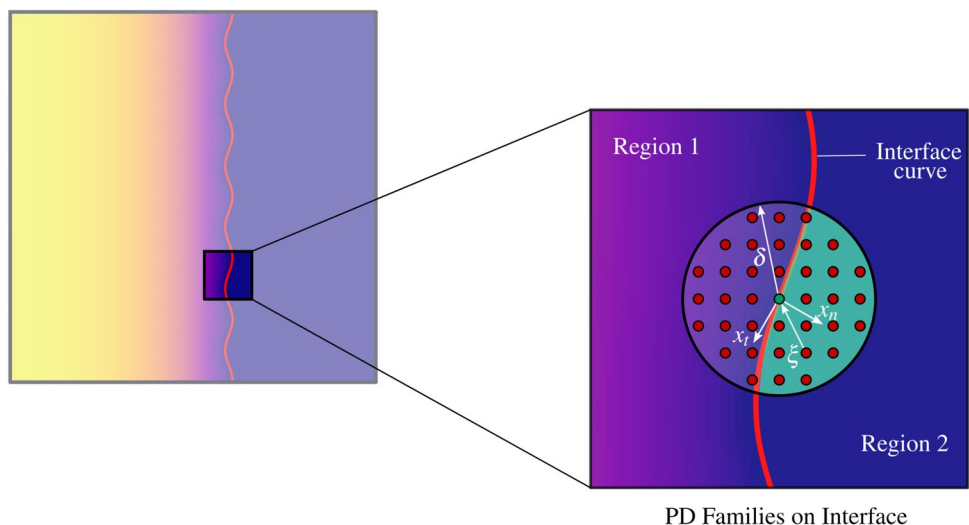


Fig. 7 Discrete points on the moving boundary (interface), rotation of the coordinate system using panel angles and calculation of the distance vector

between the coordinate locations on the interface. First, we consider a coordinate point, \tilde{p}_i^t between two neighboring points on the moving interface. Subsequently, we rotate the coordinate system based on the arctangent of the angle θ . At last, we calculate the distance vector s_i^t between the panel midpoint \tilde{p}_i^t and the nearest coordinate point on the moving interface in the next time step.

Fig. 6 Peridynamic (PD) families and interface coordinate system on the moving boundary



The norm of this distance vector can be divided to timestep size Δt to calculate the velocity of the interface in the normal direction as

$$\left[\frac{\partial \Gamma_n}{\partial t} \right]^i \approx \frac{|\mathbf{s}_i^t|}{\Delta t} \tag{15}$$

An alternative approach is to rotate the vector \mathbf{s}_i^t using the calculated panel normal and use the norm of the resulting vector as

$$\left[\frac{\partial \Gamma_n}{\partial t} \right]^i \approx \frac{|\mathbf{s}_i^t \cdot \mathbf{n}_i^t|}{\Delta t}, \tag{16}$$

where \mathbf{n}_i^t is the calculated panel normal at time t and location i . As discussed in [36], the peridynamic analysis could provide an alternative method for computing normal directions to the boundary.

Our advancement scheme of material points on the moving boundary does not prevent the material point accumulation or crossing. However, the material points can be updated if accumulation occurs since the sensor locations on the moving boundary are based on visual data.

4 Numerical results

This section presents the numerical results of the recovery experiments conducted to test the proposed learning framework. It presents the recovered models for the Stefan condition and Fisher–KPP equation, obtained for different noise levels. The relative error of the model coefficients is calculated as

$$\varepsilon_c = \frac{\|\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}\|_2}{\|\boldsymbol{\alpha}\|_2} \tag{17}$$

where $\boldsymbol{\alpha}$ is the vector of ground truth coefficients and $\hat{\boldsymbol{\alpha}}$ is the vector of the recovered coefficients. Our framework is implemented in Python, the codes and data are available at: https://github.com/alicanbekar/MB_PDDO-SINDy.

4.1 Discovery of the Stefan condition

We employ Ensemble SINDy to discover the Stefan condition from the field measurements. The candidate features consisting of the field derivatives and their products are assembled as

$$\left[1 \ u \ u_{x_n} \ u_{x_t} \ u_{x_n} \times u_{x_n} \ u_{x_n} \times u_{x_t} \ u_{x_t} \times u_{x_t} \ u_{x_n x_n} \ u_{x_t x_t} \ u_{x_n x_t} \ u_{x_n x_n} + u_{x_t x_t} \right]. \tag{18}$$

The derivatives of the field variable u are constructed in the corotational coordinate system using the PDDO as

described in detail in Appendix 1. This candidate space can also include derivatives of the field variable in the Cartesian coordinate system or higher-order derivatives and their products. In real life applications, the derivatives of the moving boundary curve with respect to the coordinate system of the field equation can also be added to the feature library, e.g. curvature of the moving boundary $\nabla^2 \Gamma$. It is worth noting that there is no guaranteed method for selecting the candidate space. However, we keep the candidate space simple and assume that it includes only the terms to explain the underlying dynamics of the moving boundary. Furthermore, we normalize each feature by its maximum absolute value to prevent the optimizer to be biased.

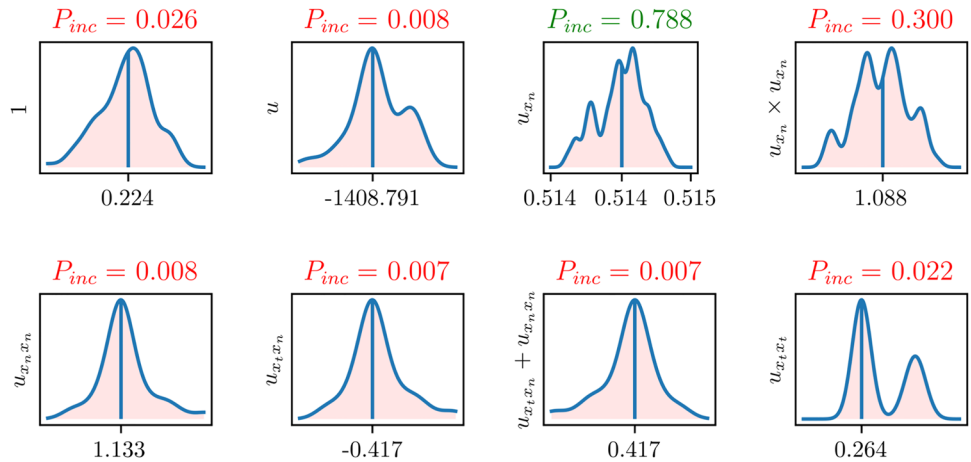
We test Ensemble SINDy with library bagging on 3 different noise levels. We aggregate the models by choosing the median of the recovered coefficients from different bootstrapped datasets. Additionally, we construct the probability distributions using Gaussian kernel estimation [37]. For the dataset with no measurement noise, we bootstrap the data for 60 times and bootstrap the features by leaving 3 out at every regression. This results in 9900 different tests. We calculate the inclusion probabilities of the features by dividing the number of appearances to the total number of tests. We set the threshold for the inclusion probability P_{inc} as 0.7. This means that we disregard the features appearing less than 70% of the time. We calculate the standard deviation of the coefficients of the features that appear more than P_{inc} , and construct the confidence intervals using the margins of 3σ from the mean μ .

We chose the parameters for the STRidge regression as $\lambda_1 = 0.3$ and $\lambda_2 = 1.0$. The selection of $\lambda_2 = 1$ is heuristic, and it is primarily used to regularize the ridge regression solutions between threshold operations. The value of λ_1 can be chosen using information criteria like BIC [38] or AIC [39]. The Python library we share contains a function to evaluate the AIC scores of the models and returns the suitable value for λ_1 .

4.1.1 Discovery of the vertical moving boundary with a sinusoidal perturbation

Figure 8 depicts the recovered coefficients, uncertainties, inclusion probabilities and the probability density estimations for the case with clean data. It shows that out of 11 terms, only 8 terms appear at least once. Also, majority of the recovered coefficients appear less than 3% of the time. Only 2 terms have significant inclusion probabilities, u_{x_n} and $u_{x_n} \times u_{x_n}$. Considering the correlation among them makes this appearance rational. The only term appearing more than the threshold probability is u_{x_n} with $P_{inc} = 0.788$ and median value of 0.514, which is the term responsible for the underlying dynamics of the moving boundary. We calculate ε_c using

Fig. 8 Median values, 99% ($\mu \pm 3\sigma$) confidence intervals and inclusion probabilities for the Stefan condition with a vertical moving boundary having a sinusoidal perturbation for the clean dataset



Eq. (17) as 0.028 which corresponds to less than 3% relative error in the recovered coefficient.

Figure 8 also shows the 99% or 3σ confidence interval of the recovered coefficient of the feature u_{x_n} . We use the limits of the confidence interval and solve for the moving boundary locations to show the uncertainty of the position of the interface. Figure 9 depicts the region of uncertainty, ground truth solution and solution obtained with the median value from the recovery. We also provide the Fréchet distance between the ground truth and predictions in Appendix 4. Since the dataset is clean, the recovered coefficients and the position of the moving boundary has a narrow region of uncertainty. Furthermore, the solution obtained with the median value of the recovery coincides with the ground truth solution.

Subsequently, we add noise to the training data as $\bar{u} = u + n$ where $n \sim \mathcal{N}(0, \sigma^2)$ and $\mathcal{N}(\mu, \sigma^2)$ is Gaussian random variable defined as $f(x) = e^{-(x-\mu)^2/2\sigma^2} / \sigma\sqrt{2\pi}$ with σ and μ representing the standard deviation and mean of the distribution, respectively, where σ is chosen to be the

standard deviation of u . The noise magnitude η multiplied with σ to introduce the desired level of noise to the dataset. The data is corrupted by considering 1% Gaussian random noise for the first trial. The learning and bragging parameters are the same as those of the clean data case.

Figure 10 depicts that adding noise increases the probability of irrelevant features appearing in the recovered model. Also, the confidence interval of the coefficient of the feature u_{x_n} increases and 3σ confidence interval corresponds to [0.316, 0.617]. Interestingly, inclusion probability of the correct model also increases with $P_{inc} = 0.975$ and median value of 0.504 is closer to the ground truth value compared to the case with clean data ($\epsilon_c = 0.008$). This is caused by the probabilistic nature of the bagging process. However, noise in the field data increases the uncertainty of the recovered model, making the recovered median value less reliable. Figure 11 depicts that the noise in the dataset widens the 99% confidence interval and increases the uncertainty of the predicted position of the moving boundary. However, the recovered position using

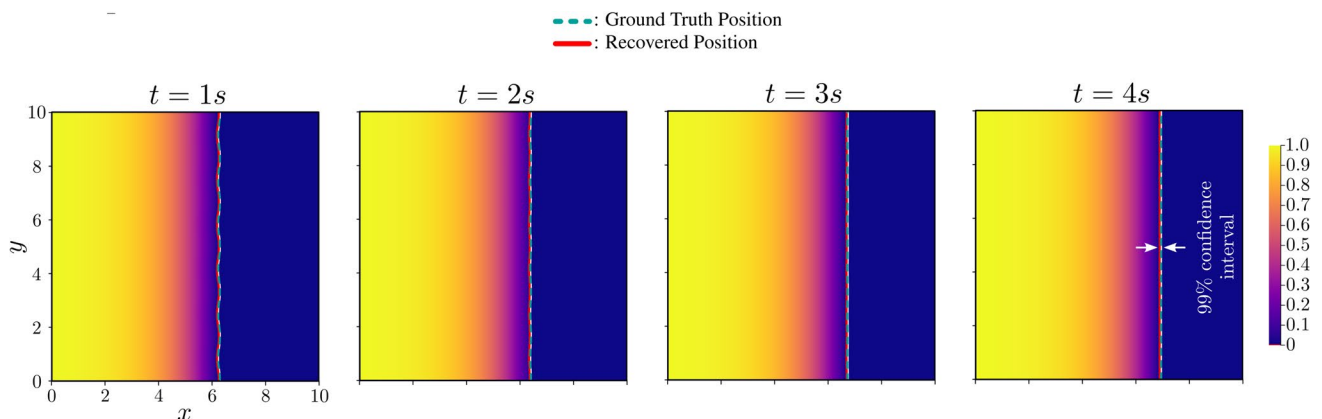
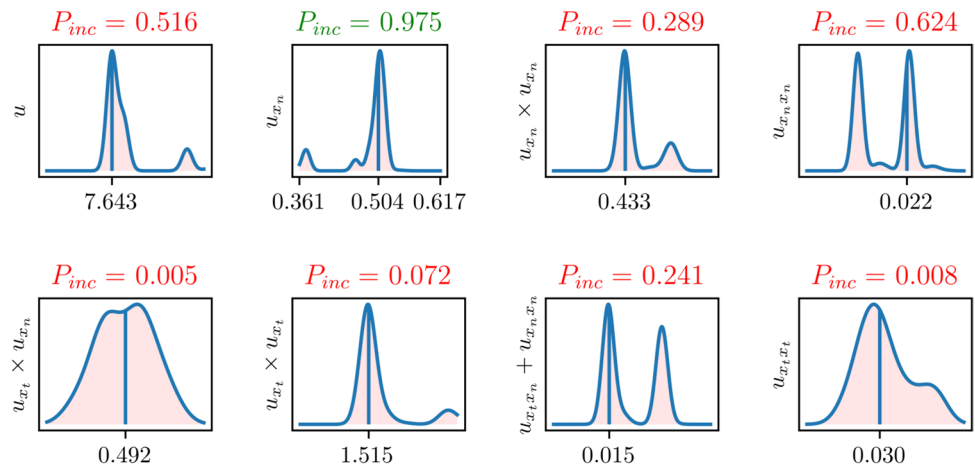


Fig. 9 Interface positions with the confidence intervals at $t = 1, 2, 3, 4$ s for the clean data case with a vertical moving boundary having a sinusoidal perturbation

Fig. 10 Median values, 99% ($\mu \pm 3\sigma$) confidence intervals and inclusion probabilities for the Stefan condition with a vertical moving boundary having a sinusoidal perturbation using dataset with 1% noise level



---: Ground Truth Position
 ---: Recovered Position

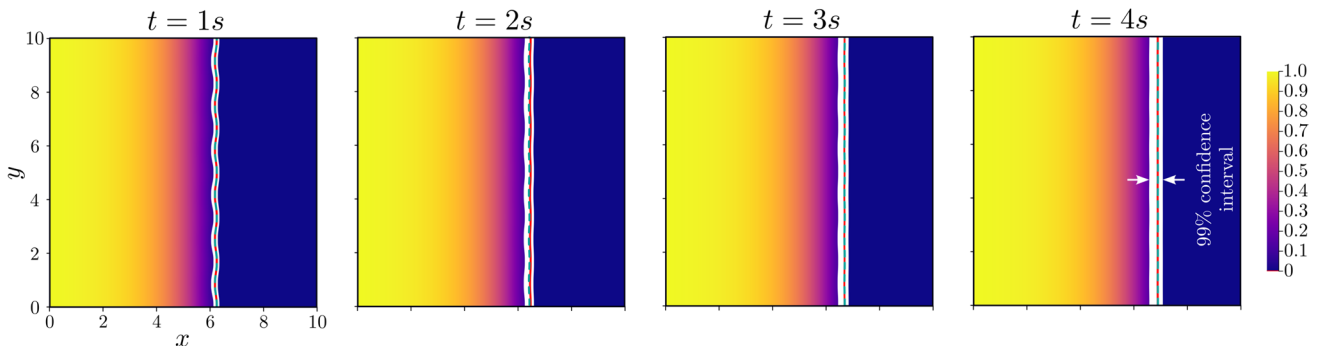


Fig. 11 Interface positions with the confidence intervals at $t = 1, 2, 3, 4$ s for the case of 1% Gaussian noise case with a vertical moving boundary having a sinusoidal perturbation

the recovered coefficient still overlaps with the ground truth position at $t = 4s$.

Finally, we corrupt the data by considering 5% Gaussian random noise. The learning and bragging parameters are the same as those of the clean data case. Figure 12 shows the results for the recovery. The 3σ confidence interval of the coefficient of the feature u_{x_n} corresponds to $[-0.054, 0.732]$ with the median value of 0.387 ($\epsilon_c = 0.226$) which indicates a larger error. Nonetheless, Ensemble SINDy can properly identify the responsible terms for underlying dynamics even in the presence of relatively high measurement noise.

Figure 13 illustrates an increasing discrepancy between the ground truth position and the recovered position as noise levels escalate. However, compared to the confidence region, recovered position is relatively close to the ground truth position. Table 1 in Appendix 2 shows the comparison of the recovered coefficients with the ground truth model.

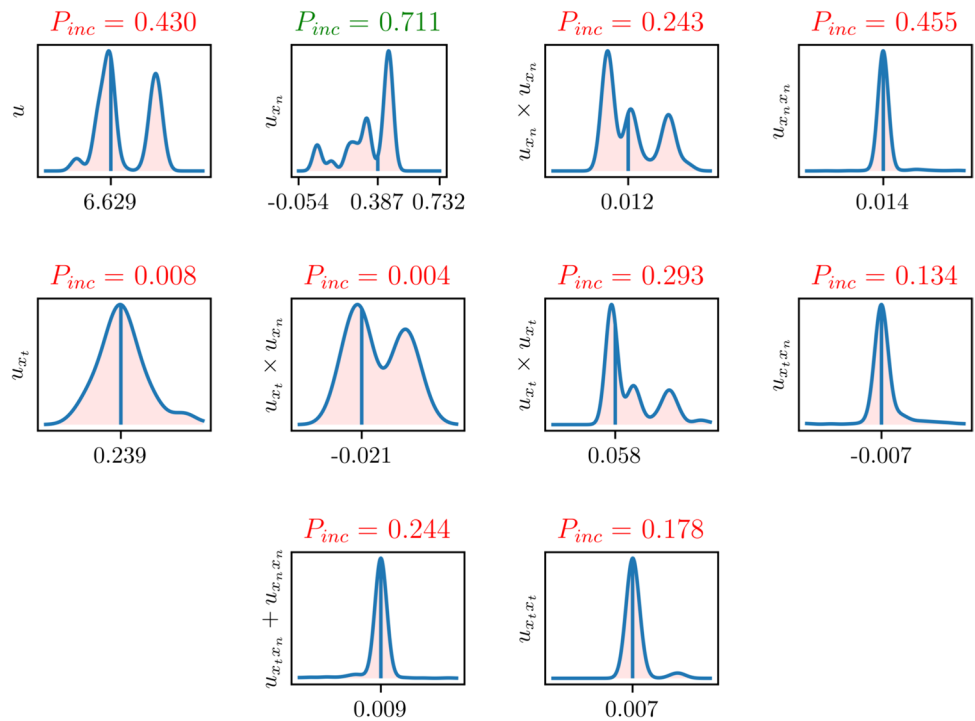
4.1.2 Discovery for circular moving boundary with irregularity

For this case, we modify the sparsity penalty and change it from $\lambda_1 = 0.3$ to $\lambda_1 = 0.1$. The reason for this modification is that none of the inclusion probabilities exceed the threshold inclusion probability for $\lambda_1 = 0.3$. Figure 14 depicts the results of the ensemble process for the clean dataset.

Figure 14 shows that u_{x_n} is the only relevant term explaining the underlying dynamics of the moving boundary, with $P_{inc} = 0.714$ and median value of 0.086 ($\epsilon_c = 0.14$). It can be seen that compared to Fig. 9, Fig. 15 shows a broader region of uncertainty. This is primarily caused by the longer duration of the analysis $t = 40$.

Subsequently, we corrupt the data using 1% Gaussian random noise. Figure 16 shows that added noise widens the confidence interval to $[0.037, 0.166]$. However, the median value of 0.105 is closer to the ground truth value compared to the case with the clean data ($\epsilon_c = 0.05$).

Fig. 12 Median values, 99% ($\mu \pm 3\sigma$) confidence intervals and inclusion probabilities for the Stefan condition with a vertical moving boundary having a sinusoidal perturbation using dataset with 5% noise level



---: Ground Truth Position
 ---: Recovered Position

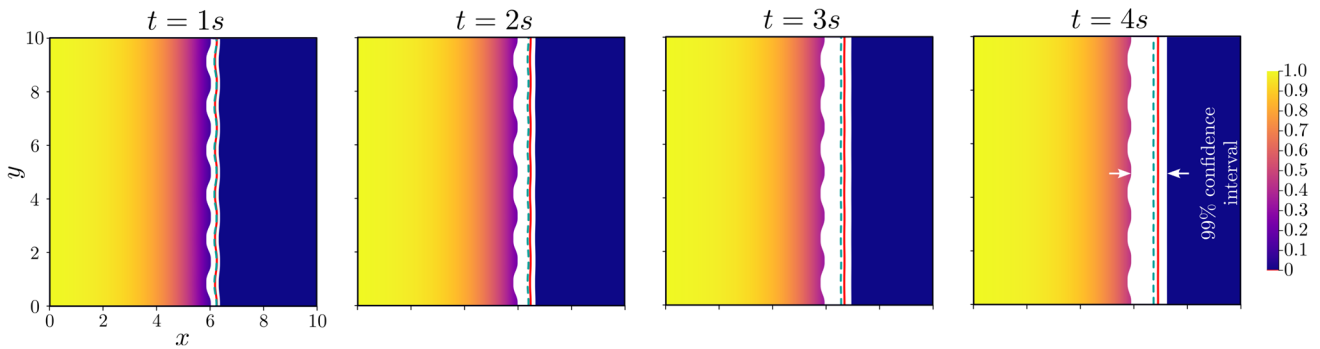


Fig. 13 Interface positions with the confidence intervals at $t = 1, 2, 3, 4$ s for the 5% Gaussian noise case with a vertical moving boundary having a sinusoidal perturbation

Fig. 14 Median values, 99% ($\mu \pm 3\sigma$) confidence intervals and inclusion probabilities for the Stefan condition with a circular moving boundary having irregularity using the clean dataset

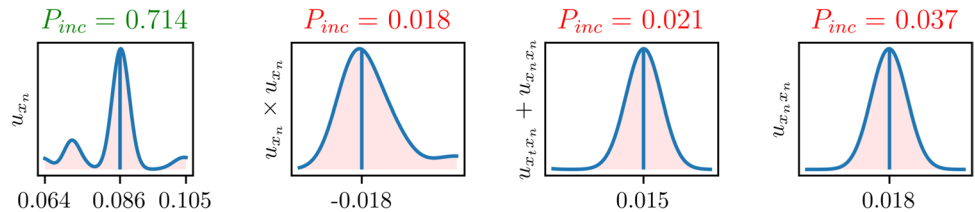


Figure 17 shows that recovered position using the median value of the recovered coefficient still overlaps with the ground truth position at $t = 40$ despite the added noise.

Finally, we corrupt the data using 5% Gaussian random noise. Figure 18 shows that similar to the clean data case of the vertical moving boundary having a sinusoidal

perturbation, the term $u_{x_n} \times u_{x_n}$ appears in addition to the term u_{x_n} . The 3σ confidence interval of the coefficient of the feature u_{x_n} corresponds to $[0.048, 0.204]$ with the median value of 0.106 ($\epsilon_c = 0.06$). Figure 19 shows the recovered position using the median value of recovered coefficients.

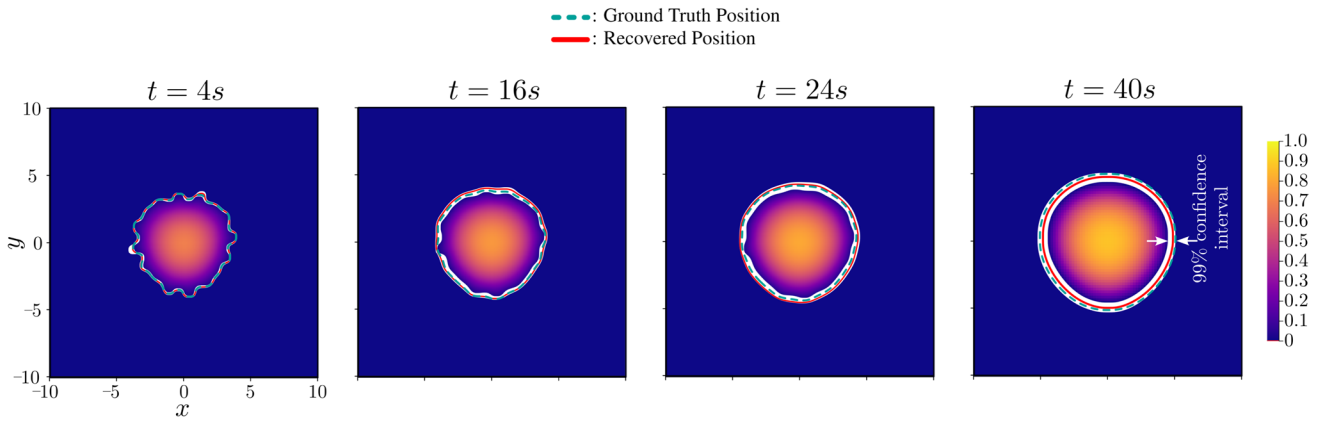


Fig. 15 Interface positions with the confidence intervals at $t = 4, 16, 24, 40$ s for the clean data case with a circular moving boundary having irregularity

Fig. 16 Median values, 99% ($\mu \pm 3\sigma$) confidence intervals and inclusion probabilities for the Stefan condition with a circular moving boundary having irregularity using dataset with 1% noise level

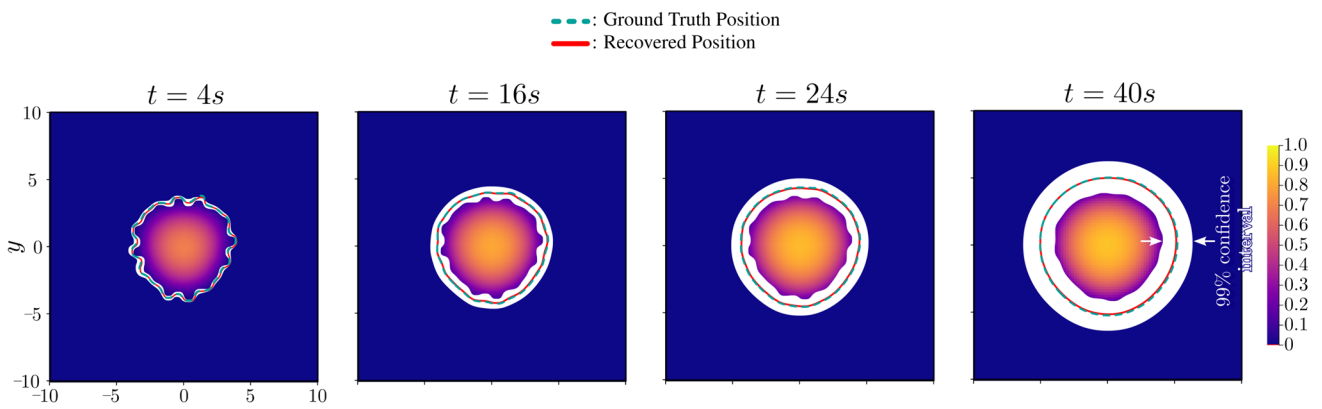
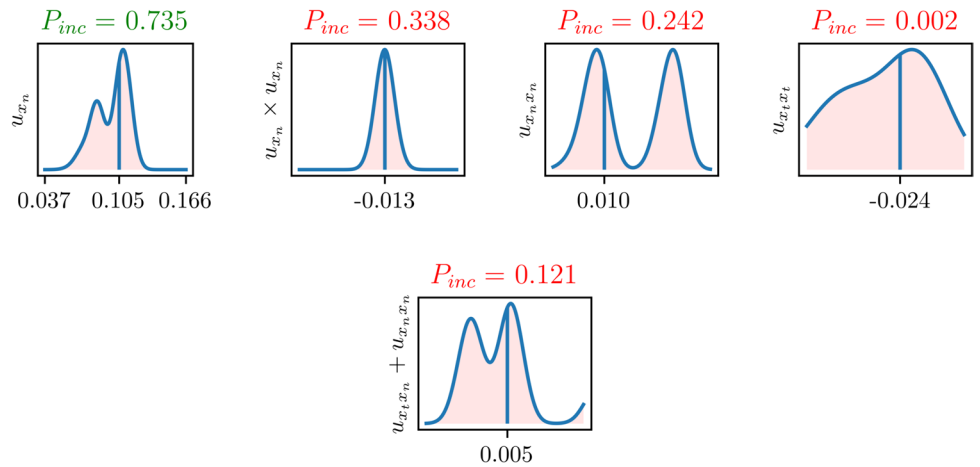
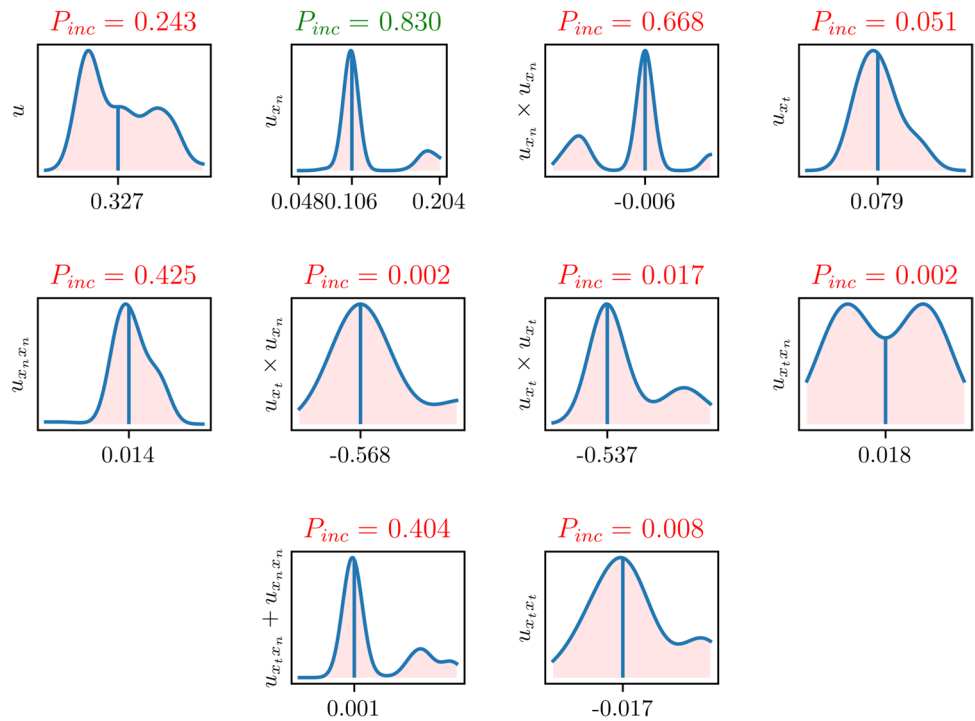


Fig. 17 Interface positions with the confidence intervals at $t = 4, 16, 24, 40$ s for the 1% Gaussian noise case with a circular moving boundary having irregularity

Fig. 18 Median values, 99% ($\mu \pm 3\sigma$) confidence intervals and inclusion probabilities for the Stefan condition with a circular moving boundary having irregularity using dataset with 5% noise level



---: Ground Truth Position
 ---: Recovered Position

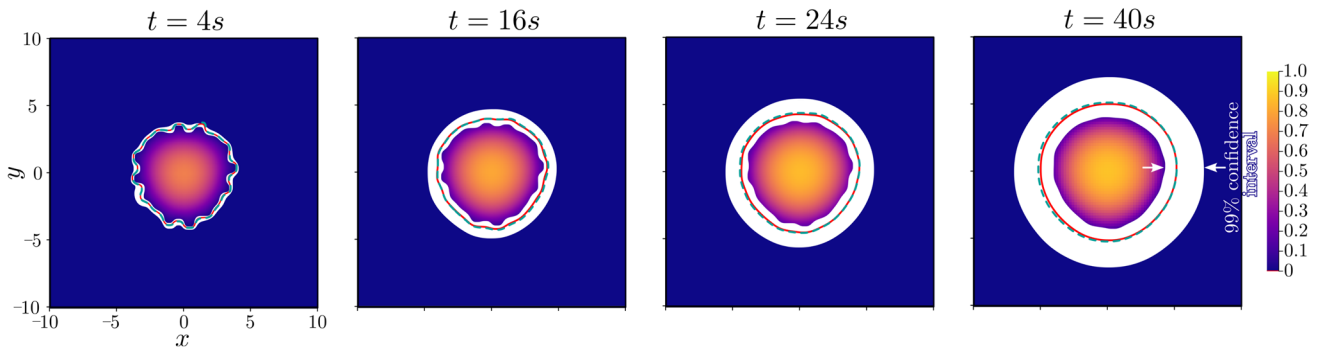


Fig. 19 Interface positions with the confidence intervals at $t = 4, 16, 24, 40$ s for the 5% Gaussian noise case with a circular moving boundary having irregularity

Table 2 in Appendix 2 shows the comparison of the recovered coefficients with the ground truth model.

4.2 Discovery of the Fisher–KPP Equation

For this case, we also employ Ensemble SINDy to discover the Fisher–KPP equation from the field measurements. Our candidate space terms for the discovery are based on the assumption that the population density evolves in a Cartesian coordinate system. Therefore, the derivatives of the field variable u are constructed in x and y directions.

This is the inductive bias of our model for the Fisher–KPP equation. The candidate space consisting the field derivatives and their products is assembled as

$$\left[1 \ u \ u^2 \ \nabla^2 u \ u_x \ u_y \ u_{xx} \ u_{xy} \ u_{yy} \ u_y u_{xx} \right]. \tag{19}$$

Again, we keep the candidate space simple because of the computational burden of the ensembling process and assume that it includes only the terms to explain the underlying dynamics of the Fisher–KPP equation. We scale each feature by its maximum absolute value.

Fisher–KPP equation with the chosen initial conditions is a challenging case for discovery. The main reason for this is that the saturation of the concentration values far from the moving boundary. When the field variable saturates, its derivatives vanish, giving no information about the field equation; nevertheless, we use Ensemble SINDy with library bagging on Fisher–KPP equation to discover the underlying dynamics. For the dataset with no measurement noise, we bootstrap the data for 60 times and bootstrap the features by leaving 1 out at every regression. This results in 600 different tests. We set the threshold for the inclusion probability P_{inc} as 0.7. We calculate the standard deviation of the coefficients of the features appearing more than P_{inc} and construct the confidence interval using 3σ . The learning and parameters are specified as the same as those of the Stefan condition experiments. Figure 20 shows the results for the recovery for Fisher–KPP equation.

Figure 20 shows that the terms u and u^2 appear with $P_{inc} = 0.889$ and median values have the same magnitude ± 0.980 with opposite signs. This means that these terms appear as a pair and can be combined to a form $u(1 - u)$. Laplacian term $\nabla^2 u$ appears with $P_{inc} = 0.778$ which is also greater than the threshold probability with the median value 0.972. Other appearing terms u_{xx} and u_{yy} also show up in pairs when Laplacian term is singled out, which is combined to form the Laplacian term. Hence, all recovered coefficients are relevant but the algorithm chooses the parsimonious solution. We calculate ϵ_c as 0.022 which corresponds to less than 3% relative error in the recovered coefficients. Figure 21 shows the absolute error between the ground truth solution and the recovered solution with the largest error being $\approx 4 \times 10^{-3}$.

Additional Gaussian noise in Fisher–KPP equation results in recovering incorrect models, vanishing derivatives might be the real cause of this failure. A potential solution to this issue can be the initialization of the same model multiple times with varying initial conditions; thereby generating a

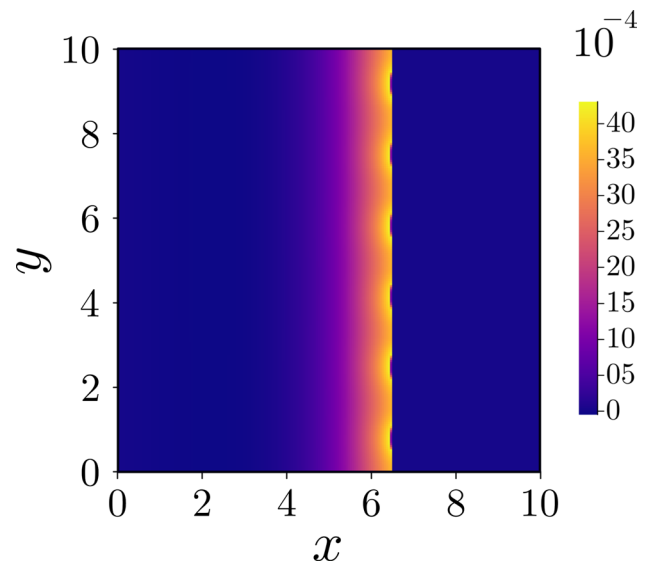


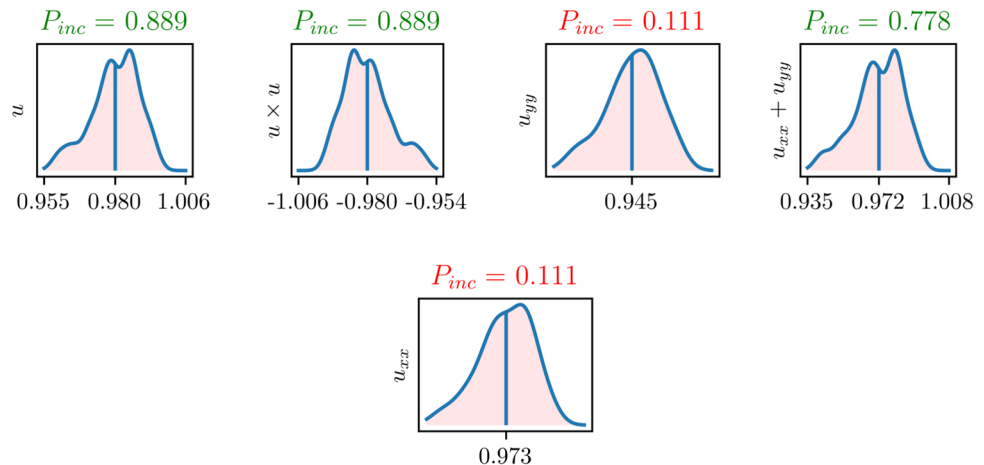
Fig. 21 The absolute error between the ground truth and the recovered solutions

more diverse training dataset. Additionally Weak SINDy can be employed for this problem to increase recovery success. However, Weak SINDy is beyond the scope of this study.

5 Discussion and conclusions

This study proposes a novel framework for learning underlying physics of processes with moving boundaries. By combining the recently introduced Ensemble SINDy and PDDO, we have successfully recovered the moving boundary equation and Fisher–KPP equation of Fisher–Stefan model. We impose the inductive bias assuming that the corotational coordinate system admits the parsimonious dynamical model. While this inductive bias does not limit the additional features evolving in a Cartesian coordinate system, it

Fig. 20 Median values, 99% ($\mu \pm 3\sigma$) confidence intervals and inclusion probabilities for the Fisher–KPP equation using the clean dataset



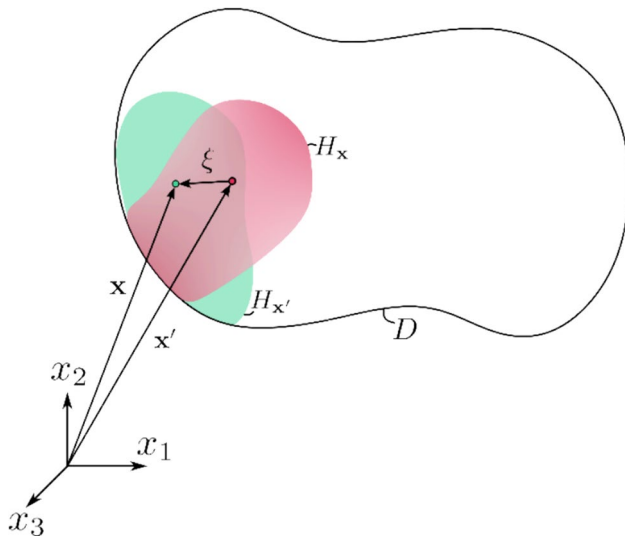


Fig. 22 Interaction of PD points, \mathbf{x} and \mathbf{x}' with arbitrary family size and shape

is a critical part of the algorithm. We have demonstrated the robustness of the present approach by considering various levels of noise in the measured data. We present the confidence intervals of recovered coefficients and demonstrate the uncertainties of the moving boundary positions by obtaining the solutions using the recovered coefficients.

Although the main focus is Fisher–Stefan model, this proposed framework is applicable to any kind of moving boundary problem with a smooth interface without a transitional region between the interacting domains. There are many other potential applications of the proposed framework, but one interesting application can be the discovery of mathematical models for regrowing limbs and tissue [40, 41]. Regrowing limbs by nature is a moving boundary problem, and mathematical understanding of this complex mechanism can help researchers govern or predict the process better. Exploring the physics of crystal growth presents another valuable application. Understanding these dynamics is crucial for enhancing the efficiency and quality of semiconductor production.

In spite of the success of the proposed framework for learning physics with moving boundaries, there is still room for improvement. One limitation is locating the moving boundary. We assume that the location of the moving boundary is known beforehand. This process can be automated using a segmentation algorithm. Another improvement is to use the suitable measurement techniques for moving boundary problems because field measurements can disturb the moving boundary. One suitable way can be digital particle image thermometry/velocimetry [42] for measuring the temperature change while following the moving boundary using the visual input. An additional feature to the proposed algorithm can be discovery of parametric PDEs. Ultimately,

proposed algorithm might also be integrated with the Weak SINDy approach, this can allow the recovery of Fisher–KPP equation in presence of noise. A possible extension of the Weak SINDy can be by using the weak form of peridynamics for discovering the nonlocal forms of the local PDEs [42]. The assumption of the parsimonious dynamics evolve in moving boundary curve corotational coordinate system can also be relaxed using an approach similar to [18].

Appendix 1: Peridynamic differential operator

PDDO is based on the idea of peridynamic interactions which was originally proposed by Silling and Lehoucq [44] in his work about Peridynamic (PD) theory. With the PDDO, the nonlocal representation of a field $f = f(\mathbf{x})$ can be constructed at a point \mathbf{x} by incorporating the effect of its interaction with the other points, \mathbf{x}' in its vicinity. The extent of this interaction is limited to a finite radius called the horizon radius, δ . The unique set of points interacting with the point of interest is called the family of point of interest. This family can be mathematically described as $\mathbf{H}_x = \{ \mathbf{x}' | d(\mathbf{x}' - \mathbf{x}) < \delta \}$. Each point inside the family occupies an infinitesimal volume area or length depending on the dimensionality of the domain. Similarly, all family members \mathbf{x}' also have their own families, $\mathbf{H}_{x'}$. Moreover, there is no constraint on families being symmetric. The relative position of family members to the point of interest is defined as $\xi = \mathbf{x}' - \mathbf{x}$. Figure 22 depicts the families and interactions between material points.

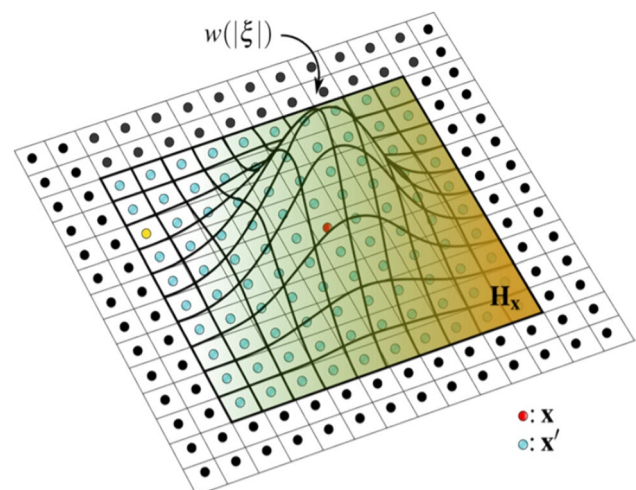


Fig. 23 Interaction domain \mathbf{H}_x of point \mathbf{x} and the form of the Gaussian weight function

In a 2-dimensional space, a function, $f(\mathbf{x} + \boldsymbol{\xi})$ can be expressed in terms of Taylor Series Expansion (TSE) as

$$\begin{aligned}
 f(\mathbf{x} + \boldsymbol{\xi}) = & f(\mathbf{x}) + \xi_1 \frac{\partial f(\mathbf{x})}{\partial x_1} + \xi_2 \frac{\partial f(\mathbf{x})}{\partial x_2} \\
 & + \frac{1}{2!} \xi_1^2 \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} + \frac{1}{2!} \xi_2^2 \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} \\
 & + \xi_1 \xi_2 \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} + \mathcal{R},
 \end{aligned}
 \tag{20}$$

where \mathcal{R} is the remainder. Multiplying each term with PD functions, $g_2^{p_1 p_2}(\boldsymbol{\xi})$ and integrating over the domain of interaction (family), \mathbf{H}_x result in

$$\begin{aligned}
 & \int_{\mathbf{H}_x} f(\mathbf{x} + \boldsymbol{\xi}) g_2^{p_1 p_2}(\boldsymbol{\xi}) dV \\
 = & \int_{\mathbf{H}_x} f(\mathbf{x}) g_2^{p_1 p_2}(\boldsymbol{\xi}) dV + \frac{\partial f(\mathbf{x})}{\partial x_1} \\
 & \int_{\mathbf{H}_x} \xi_1 g_2^{p_1 p_2}(\boldsymbol{\xi}) dV + \frac{\partial f(\mathbf{x})}{\partial x_2} \\
 & \int_{\mathbf{H}_x} \xi_2 g_2^{p_1 p_2}(\boldsymbol{\xi}) dV \\
 & + \frac{1}{2} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} \\
 & \int_{\mathbf{H}_x} \xi_1^2 g_2^{p_1 p_2}(\boldsymbol{\xi}) dV + \frac{1}{2} \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} \\
 & \int_{\mathbf{H}_x} \xi_2^2 g_2^{p_1 p_2}(\boldsymbol{\xi}) dV + \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} \\
 & \int_{\mathbf{H}_x} \xi_1 \xi_2 g_2^{p_1 p_2}(\boldsymbol{\xi}) dV,
 \end{aligned}
 \tag{21}$$

in which the point \mathbf{x} is not necessarily symmetrically located in the domain of interaction.

The initial relative position, $\boldsymbol{\xi}$, between points \mathbf{x} and \mathbf{x}' can be expressed as $\boldsymbol{\xi} = \mathbf{x}' - \mathbf{x}$. This ability permits each point to have its own unique family with an arbitrary position. Therefore, the size and shape of each family can be different, and they significantly influence the degree of nonlocality. In general, the family of a point can be non-symmetric due to nonuniform spatial discretization.

The degree of interaction between the material points in each family is specified by a nondimensional weight function, $w(|\boldsymbol{\xi}|)$, which can vary from point to point. The weight function is usually chosen as Gaussian as $w(|\boldsymbol{\xi}|) = e^{-4|\boldsymbol{\xi}|^2/\delta^2}$. This weight function and the interaction domain are shown Fig. 23.

The interactions become more local with decreasing family size. Thus, the family size and shape are important parameters. Each point occupies an infinitesimally small entity such as volume, area or a distance. The PD functions are constructed such that they are orthogonal to each term in the TSE as

$$\frac{1}{n_1! n_2!} \int_{\mathbf{H}_x} \xi_1^{n_1} \xi_2^{n_2} g_2^{p_1 p_2}(\boldsymbol{\xi}) dV = \delta_{n_1 p_1} \delta_{n_2 p_2}
 \tag{22}$$

with $(n_1, n_2, p, q = 0, 1, 2)$ and δ_{ij} is the Kronecker delta symbol. Enforcing the orthogonality conditions in the TSE leads to the nonlocal PD representation of the function itself and its derivatives as

$$f(\mathbf{x}) = \int_{\mathbf{H}_x} f(\mathbf{x} + \boldsymbol{\xi}) g_2^{00}(\boldsymbol{\xi}) dV,
 \tag{23a}$$

$$\left\{ \frac{\partial f(\mathbf{x})}{\partial x} \quad \frac{\partial f(\mathbf{x})}{\partial y} \right\} = \int_{\mathbf{H}_x} f(\mathbf{x} + \boldsymbol{\xi}) \left\{ g_2^{10}(\boldsymbol{\xi}) \quad g_2^{01}(\boldsymbol{\xi}) \right\} dV,
 \tag{23b}$$

$$\left\{ \frac{\partial^2 f(\mathbf{x})}{\partial x^2} \quad \frac{\partial^2 f(\mathbf{x})}{\partial y^2} \quad \frac{\partial^2 f(\mathbf{x})}{\partial x \partial y} \right\} = \int_{\mathbf{H}_x} f(\mathbf{x} + \boldsymbol{\xi}) \left\{ g_2^{20}(\boldsymbol{\xi}) \quad g_2^{02}(\boldsymbol{\xi}) \quad g_2^{11}(\boldsymbol{\xi}) \right\} dV.
 \tag{23c}$$

The PD functions can be constructed as a linear combination of polynomial basis functions

$$\begin{aligned}
 g_2^{p_1 p_2} = & a_{00}^{p_1 p_2} w_{00}(|\boldsymbol{\xi}|) + a_{10}^{p_1 p_2} w_{10}(|\boldsymbol{\xi}|) \xi_1 + a_{01}^{p_1 p_2} w_{01}(|\boldsymbol{\xi}|) \xi_2 + a_{20}^{p_1 p_2} w_{20}(|\boldsymbol{\xi}|) \xi_1^2 \\
 & + a_{02}^{p_1 p_2} w_{02}(|\boldsymbol{\xi}|) \xi_2^2 + a_{11}^{p_1 p_2} w_{11}(|\boldsymbol{\xi}|) \xi_1 \xi_2,
 \end{aligned}
 \tag{24}$$

where $a_{q_1 q_2}^{p_1 p_2}$ are the unknown coefficients, $w_{q_1 q_2}(|\boldsymbol{\xi}|)$ are the influence functions, and ξ_1 and ξ_2 are the components of the vector $\boldsymbol{\xi}$. Assuming $w_{q_1 q_2}(|\boldsymbol{\xi}|) = w(|\boldsymbol{\xi}|)$ and incorporating the PD functions into the orthogonality equation lead to a system of algebraic equations for the determination of the coefficients as

$$\mathbf{Aa} = \mathbf{b},
 \tag{25}$$

After determining the coefficients $a_{q_1 q_2}^{p_1 p_2}$ via $\mathbf{a} = \mathbf{A}^{-1} \mathbf{b}$, the PD functions $g_2^{p_1 p_2}(\boldsymbol{\xi})$ can be constructed. The detailed derivations and the associated computer programs can be found in [42]. The PDDO is nonlocal; however, in the limit as the horizon size approaches zero, it recovers the local differentiation as proven by Silling and Lehoucq [43]. The peridynamic model defines the nonlocality length scale using δ . On the contrary, classical theories lack parameters to encapsulate nonlocal interactions. The PD model recovers the local solution when the horizon approaches zero. It's important to note that this convergence may not necessarily be uniform. Silling and Lehoucq [43] have further illustrated that if the motion is not twice continuously differentiable or if the PD constitutive model lacks continuous differentiability, the convergence to a classical

model is not guaranteed in the scenario of a small horizon. In such a situation, while PD equations remain valid for any positive horizon, the properties of convergence at a zero-horizon limit are indeterminate.

Appendix 2: Recovery results for Stefan condition

See Tables 1 and 2.

Table 1 Identified models and corresponding confidence intervals for Stefan condition with a vertical moving boundary having a sinusoidal perturbation

	Model	Confidence Interval
Ground truth	$\frac{\partial \Gamma_n}{\partial t} = -0.5u_{x_n}$	–
Clean data	$\frac{\partial \Gamma_n}{\partial t} = -0.514u_{x_n}$	[0.514, 0.515]
Identified model (1% noise)	$\frac{\partial \Gamma_n}{\partial t} = -0.504u_{x_n}$	[0.316, 0.617]
Identified model (5% noise)	$\frac{\partial \Gamma_n}{\partial t} = -0.387u_{x_n}$	[-0.054, 0.732]

Table 2 Identified models and corresponding confidence intervals for Stefan condition with a circular moving boundary having irregularity

	Model	Confidence Interval
Ground truth	$\frac{\partial \Gamma_n}{\partial t} = -0.1u_{x_n}$	–
Clean data	$\frac{\partial \Gamma_n}{\partial t} = -0.086u_{x_n}$	[0.064, 0.105]
Identified model (1% noise)	$\frac{\partial \Gamma_n}{\partial t} = -0.105u_{x_n}$	[0.037, 0.166]
Identified model (5% noise)	$\frac{\partial \Gamma_n}{\partial t} = -0.106u_{x_n}$	[0.048, 0.204]

Appendix 3: Data generation

The solutions for the vertical moving boundary with a sinusoidal perturbation and the circular moving boundary with irregularity are obtained by using the existing software [32] and [31], respectively. Both methods replace the original set of Fisher–KPP equations with a level set formulation and solve the system using the finite difference method. For the vertical moving boundary with a sinusoidal perturbation, the initial position of the moving boundary is set at $x = 6$ with a sinusoidal perturbation of amplitude $\varepsilon = 0.1$ and wavelength $p = 2$. We consider all snapshots from the simulations. For testing, we employ 201 discretization points in each direction, including 201 grid points for the moving boundary for testing.

Appendix 4: Fréchet distances

The Fréchet distance metric can be used to measure the similarity between parametric curves [45]. In this study, we use the discrete Fréchet distance to compare the recovered

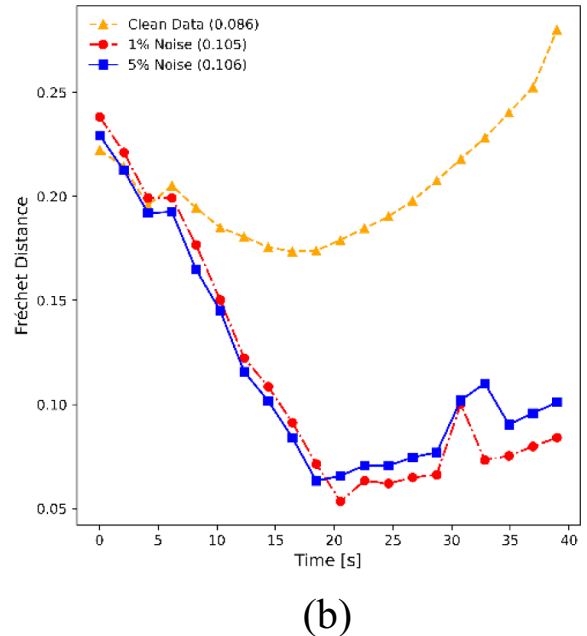
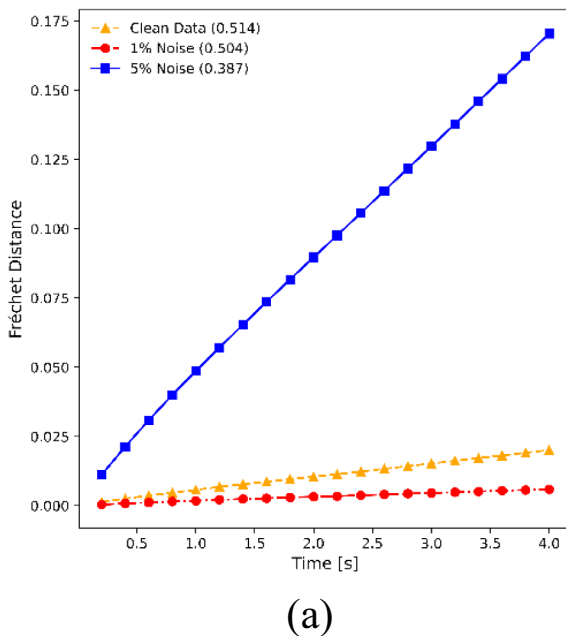


Fig. 24 Fréchet distances between the ground truth and recovered moving boundary curves: **a** vertical moving boundary with a sinusoidal perturbation, **b** circular moving boundary with irregularity

and ground truth moving boundary curves by utilizing the existing software [46]. The curve coordinates are interpolated linearly using consistent arc lengths by considering a fixed set of points. We consider 100 points for the sinusoidal perturbation case and 300 points for the irregularity case to capture the irregular curve with better accuracy. We then calculate the discrete Fréchet distance between the ground truth and predicted curves using the recovered coefficients.

Figure 24 illustrates the discrepancy and error accumulation between the ground truth and predicted curves. In the first case shown in Fig. 24a, where time intervals are smaller and the moving boundary curve is easier to predict, we observe a gradually increasing error for the recovered cases. However, for the circular moving boundary, the time intervals are larger, and the curve is more complex. This complexity causes the shape of the curve to change rapidly at the beginning of the simulation, leading to a greater distance between the predicted and ground truth curves from the first timestep. As the simulation progresses, the Stefan condition smooths out the irregularities, resulting in the predicted curve approaching the ground truth. Nevertheless, in the case of clean data, the distance between the smoothed curves increases over time, which explains the increasing Fréchet distance between the predicted and ground truth curves.

The discontinuities observed in Fig. 24b could be due to the large timesteps used in the simulation or might be numerical artifacts from the interpolation process. However, these discontinuities still effectively illustrate the trend of error accumulation and the similarity between the predicted and ground truth curves.

Acknowledgements EM and ACB performed this work as part of the ongoing research at the MURI Center for Material Failure Prediction through Peridynamics at the University of Arizona (AFOSR Grant No. FA9550-14-1-0073).

Author contribution A.C.B.: Conceptualization, Methodology, Analysis, Software, Original draft E.H.: Conceptualization, Methodology, Review, Editing E.M.: Conceptualization, Methodology, Review, Editing.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

- Crowley AB (1978) Numerical solution of stefan problems. *Int J Heat Mass Transf* 21:215–219. [https://doi.org/10.1016/0017-9310\(78\)90225-9](https://doi.org/10.1016/0017-9310(78)90225-9)
- Dalwadi MP, Waters SL, Byrne HM, Hewitt IJ (2020) A mathematical framework for developing freezing protocols in the cryopreservation of cells. *SIAM J Appl Math* 80:657–689. <https://doi.org/10.1137/19m1275875>
- Friedman A (2015) Free boundary problems in biology. *Philos Trans R Soc A Math Phys Eng Sci.* <https://doi.org/10.1098/RSTA.2014.0368>
- Stansby PK, Chegini A, Barnes TCD (1998) The initial stages of dam-break flow. *J Fluid Mech* 374:407–424. <https://doi.org/10.1017/s0022112098001918>
- Friedman A (2000) Free boundary problems in science and technology. *Not AMS* 47:854–861
- Scardovelli R, Zaleski S (2003) Direct numerical simulation of free-surface and interfacial flow. *Annu Rev Fluid Mech* 31:567–603. <https://doi.org/10.1146/annurev.fluid.31.1.567>
- Sethian JA, Smereka P (2003) Level set methods for fluid interfaces. *Annu Rev Fluid Mech* 35:341–372. <https://doi.org/10.1146/annurev.fluid.35.101101.161105>
- Gupta SC (2017) The classical Stefan problem: basic concepts, modelling and analysis with quasi-analytical solutions and methods. Elsevier, Amsterdam
- Date AW (1992) Novel strongly implicit enthalpy formulation for multidimensional stefan problems. *Numer Heat Transf Part B Fundam* 21:231–251. <https://doi.org/10.1080/10407799208944918>
- Chen S, Merriman B, Osher S, Smereka P (1997) A simple level set method for solving stefan problems. *J Comput Phys* 135:8–29. <https://doi.org/10.1006/jcph.1997.5721>
- Wang S, Perdikaris P (2021) Deep learning of free boundary and Stefan problems. *J Comput Phys* 428:109914. <https://doi.org/10.1016/j.jcp.2020.109914>
- Schmid PJ (2010) Dynamic mode decomposition of numerical and experimental data. *J Fluid Mech* 656:5–28. <https://doi.org/10.1017/s0022112010001217>
- Jia X, Willard J, Karpatne A, Read JS, Zwart JA, Steinbach M, Kumar V (2021) Physics-guided machine learning for scientific discovery: an application in simulating lake temperature profiles. *ACM/IMS Trans Data Sci* 2:1–26. <https://doi.org/10.1145/3447814>
- Bongard J, Lipson H (2007) Automated reverse engineering of nonlinear dynamical systems. *Proc Natl Acad Sci* 104:9943–9948. <https://doi.org/10.1073/pnas.0609476104>
- Schaeffer H (2017) Learning partial differential equations via data discovery and sparse optimization. *Proc R Soc A Math Phys Eng Sci.* <https://doi.org/10.1098/rspa.2016.0446>
- Brunton SL, Proctor JL, Kutz JN (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc Natl Acad Sci* 113:3932–3937. <https://doi.org/10.1073/pnas.1517384113>
- Zhang L, Schaeffer H (2019) On the convergence of the sindy algorithm. *Multiscale Model Simul* 17:948–972. <https://doi.org/10.1137/18M1189828>
- Champion K, Lusch B, Nathan Kutz J, Brunton SL (2019) Data-driven discovery of coordinates and governing equations. *Proc Natl Acad Sci USA* 116:22445–22451. <https://doi.org/10.1073/pnas.1906995116>
- Kaiser E, Kutz N, Brunton SL, Desantis D, Wolfram PJ, Bennett K, Kaheman K (2022) Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data. *Mach Learn Sci Technol* 3:015031. <https://doi.org/10.1088/2632-2153/AC567A>
- Rudy S, Alla A, Brunton SL, Nathan Kutz J (2019) Data-driven identification of parametric partial differential equations. *SIAM J Appl Dyn Syst* 18:643–660. <https://doi.org/10.1137/18M1191944>
- Shea DE, Brunton SL, Kutz JN (2020) SINDy-BVP: sparse identification of nonlinear dynamics for boundary value problems. *Phys Rev Res.* <https://doi.org/10.1103/physrevresearch.3.023255>

22. Schaeffer H, McCalla SG (2017) Sparse model selection via integral terms. *Phys Rev E* 96:023302. <https://doi.org/10.1103/PhysRevE.96.023302>
23. Messenger DA, Bortz DM (2021) Weak SINDy: Galerkin-based data-driven model selection. *Multiscale Model Simul* 19:1474–1497. <https://doi.org/10.1137/20M1343166>
24. Messenger DA, Bortz DM (2021) Weak SINDy for partial differential equations. *J Comput Phys* 443:110525. <https://doi.org/10.1016/j.jcp.2021.110525>
25. Fasel U, Kutz JN, Brunton BW, Brunton SL (2022) Ensemble-SINDy: robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proc R Soc A Math Phys Eng Sci*. <https://doi.org/10.1098/rspa.2021.0904>
26. Madenci E, Barut A, Futch M (2016) Peridynamic differential operator and its applications. *Comput Methods Appl Mech Eng* 304:408–451. <https://doi.org/10.1016/j.cma.2016.02.028>
27. Bekar AC, Madenci E (2021) Peridynamics enabled learning partial differential equations. *J Comput Phys*. <https://doi.org/10.1016/j.jcp.2021.110193>
28. Rudy SH, Brunton SL, Proctor JL, Kutz JN (2017) Data-driven discovery of partial differential equations. *Sci Adv* 3:e1602614. <https://doi.org/10.1126/sciadv.1602614>
29. Breiman L (1996) Bagging predictors. *Mach Learn* 24(2):123–140. <https://doi.org/10.1007/BF00058655>
30. Du Y, Lin Z (2010) Spreading-vanishing dichotomy in the diffusive logistic model with a free boundary. *SIAM J Math Anal* 42:377–405. <https://doi.org/10.1137/090771089>
31. Tam AKY, Simpson MJ (2022) The effect of geometry on survival and extinction in a moving-boundary problem motivated by the Fisher–KPP equation. *Phys D Nonlinear Phenom* 438:133305. <https://doi.org/10.1016/J.PHYSD.2022.133305>
32. Tam AKY, Simpson MJ (2023) Pattern formation and front stability for a moving-boundary model of biological invasion and recession. *Phys D Nonlinear Phenom* 444:133593. <https://doi.org/10.1016/J.PHYSD.2022.133593>
33. Witelski TP (1995) Merging traveling waves for the porous-Fisher's equation. *Appl Math Lett* 8:57–62. [https://doi.org/10.1016/0893-9659\(95\)00047-T](https://doi.org/10.1016/0893-9659(95)00047-T)
34. Du Y, Guo Z (2012) The Stefan problem for the Fisher–KPP equation. *J Differ Equ* 253:996–1035. <https://doi.org/10.1016/J.JDE.2012.04.014>
35. Crank J, Crank J (1984) Free and moving boundary problems. Oxford University Press, New York
36. Zhao J, Jafarzadeh S, Chen Z, Bobaru F (2024) Enforcing local boundary conditions in peridynamic models of diffusion with singularities and on arbitrary domains. *Eng Comput*. <https://doi.org/10.1007/s00366-024-01995-z>
37. Scott DW (2015) Multivariate density estimation: theory, practice, and visualization. Wiley, Hoboken
38. Schwarz G (1978) Estimating the dimension of a model. *Ann Stat* 6:461–464. <https://doi.org/10.1214/aos/1176344136>
39. Akaike H (1974) A new look at the statistical model identification. *IEEE Trans Autom Control* 19:716–723. <https://doi.org/10.1109/TAC.1974.1100705>
40. Abrams MJ, Basinger T, Yuan W, Guo C-L, Goentoro L (2015) Self-repairing symmetry in jellyfish through mechanically driven reorganization. *Proc Natl Acad Sci* 112:E3365–E3373. <https://doi.org/10.1073/pnas.1502497112>
41. Abrams MJ, Tan F, Li Y, Basinger T, Heithe ML, Sarma AA, Lee IT, Conditte ZJ, Raffiee M, Dabiri JO, Gold DA, Goentoro L (2021) A conserved strategy for inducing appendage regeneration in moon jellyfish, *Drosophila*, and mice. *Elife*. <https://doi.org/10.7554/ELIFE.65092>
42. Dabiri D (2009) Digital particle image thermometry/velocimetry: a review. *Exp Fluids* 46:191–241
43. Madenci E, Barut A, Dorduncu M (2019) Peridynamic differential operator for numerical analysis. Springer, Cham
44. Silling SA, Lehoucq RB (2010) Peridynamic theory of solid mechanics. *Adv Appl Mech*. [https://doi.org/10.1016/S0065-2156\(10\)44002-8](https://doi.org/10.1016/S0065-2156(10)44002-8)
45. Vodolazskiy E (2023) Discrete Fréchet distance for closed curves. *Compu Geom Theory Appl* 111:101967. <https://doi.org/10.1016/j.comgeo.2022.101967>
46. Spiros D, Pikoula M (2019) spiros/discrete_frechet: meerkat stable release. Zenodo. <https://doi.org/10.5281/zenodo.3366385>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.