mathematics

MDPI

Article

Discovering PDEs Corrections from Data Within a Hybrid Modeling Framework

Chady Ghnatios [1,*] and Francisco Chinesta [2,3]

[1] Department of Mechanical Engineering, University of North Florida, Jacksonville, FL 32224, USA
[2] Process and Engineering in Mechanics and Materials Laboratory, UMR CNRS—Arts et Metiers Institute of Technology, 75013 Paris, France; francisco.chinesta@ensam.eu
[3] CNRS@CREATE, Create Way, 08-01 CREATE Tower, Singapore 138602, Singapore
* Correspondence: chady.ghnatios@unf.edu

Abstract: In the context of hybrid twins, a data-driven enrichment is added to the physics-based solution to represent with higher accuracy the reference solution assumed to be known at different points in the physical domain. Such an approach enables better predictions. However, the data-driven enrichment is usually represented by a regression, whose main drawbacks are (i) the difficulty of understanding the subjacent physics and (ii) the risks induced by the data-driven model extrapolation. This paper proposes a procedure enabling the extraction of a differential operator associated with the enrichment provided by the data-driven regression. For that purpose, a sparse Singular Value Decomposition, SVD, is introduced. It is then employed, first, in a full operator representation regularized optimization problem, where sparsity is promoted, leading to a linear programming problem, and then in a tensor decomposition of the operator's identification procedure. The results show the ability of the method to identify the exact missing operators from the model. The regularized optimization problem was also able to identify the weights of the missing terms with a relative error of about 10% on average, depending on the selected use case.

Keywords: scientific machine learning; model discovery; physics-informed neural networks; PINN; sparse SVD

MSC: 65D15; 65D40

check for updates

Academic Editor: Giovanni Stabile

Received: 19 November 2024
Revised: 15 December 2024
Accepted: 20 December 2024
Published: 24 December 2024

Citation: Ghnatios, C.; Chinesta, F. Discovering PDEs Corrections from Data Within a Hybrid Modeling Framework. *Mathematics* **2025**, *13*, 5. https://doi.org/10.3390/math13010005

Correction Statement: This article has been republished with a minor change. The change does not affect the scientific content of the article and further details are available within the backmatter of the website version of this article.

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Physics-based models make use of different differential operators, linear or nonlinear, involving space and time. In what follows, and without loss of generality, we assume a linear problem, defined in the domain $\Omega \subset \mathbb{R}^2$. The points in Ω are noted by $\mathbf{x}$, with components (x, y). Thus, the problem can be expressed as

$$\mathcal{L}^p(u^p(\mathbf{x})) = F(\mathbf{x}), \tag{1}$$

where $\mathcal{L}^p(\cdot)$ represents the linear differential operator, the superscript $\bullet^p$, highlighting the fact that the operator $\mathcal{L}$ comes from a state-of-the-art physics-based model, $u^p(\mathbf{x})$ its associated unknown field and $F(\mathbf{x})$ the problem source term.

The solution of the problem results in the so-called physics-based model solution $u^p(\mathbf{x})$, $\forall \mathbf{x} \in \Omega$. Now, we assume that in different points $\mathbf{x}_1, \dots, \mathbf{x}_P$, the reference solution $u(\mathbf{x})$ is accessible, from appropriate and accurate measurements, offering the values $u_1, \dots, u_P$, with $u_i \equiv u(\mathbf{x}_i)$, $i = 1, \dots, P$.

Now, to evaluate the quality of the model, a usual procedure consists of comparing these data with the model-based predictions and defining the associated local gaps:

$$\begin{cases} \delta_1 = u^p(\mathbf{x}_1) - u_1, \\ \delta_2 = u^p(\mathbf{x}_2) - u_2, \\ \vdots \\ \delta_P = u^p(\mathbf{x}_P) - u_P. \end{cases} \tag{2}$$

Now, vector Δ is introduced, containing the different local gaps δ_i, $i = 1, \ldots, P$:

$$\Delta = \begin{pmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_P \end{pmatrix}. \tag{3}$$

When $\|\Delta\| \approx 0$, it can be concluded that the physics-based model $\mathcal{L}^p(\cdot)$ accurately represents the physics under scrutiny. However, when $\|\Delta\| > \epsilon$ (ϵ representing the accuracy threshold), the model fails to represent the observed reality, and a correction seems compulsory.

First, a regression $\delta_\theta(\mathbf{x})$ is constructed, where θ refers to the regression parameters. The regression can be linear (e.g., regularized polynomial sparse Proper Generalized Decomposition–sPGD) or nonlinear (e.g., a multi-layer perceptron neural network). The regression training employs the available data $(\mathbf{x}_i, \delta_i)$, $i = 1, \ldots, P$.

Thus, the usual hybrid modeling proceeds as follows:

1. Define and solve the physics-based model: $\mathcal{L}^p(u^p(\mathbf{x})) = F \rightarrow u^p(\mathbf{x})$.
2. From the available data, $(\mathbf{x}_i, \delta_i)$, $i = 1, \ldots, P$, construct the regression $\delta_\theta(\mathbf{x})$.
3. The solution could be expressed as the addition of both contributions, that is $u(\mathbf{x}) = u^p(\mathbf{x}) + \delta_\theta(\mathbf{x})$, where for the sake of notational simplicity the correction term will be noted by $u^c(\mathbf{x})$, referring the field correction, emphasized by the superscript $\bullet^c$:

$$u(\mathbf{x}) = u^p(\mathbf{x}) + u^c(\mathbf{x}). \tag{4}$$

Even if knowing $u^c(\mathbf{x})$ offers a clear advantage to improve the prediction accuracy, its nature remains quite elusive, much less intuitive than the physics-based contribution. In fact, $u^p(\mathbf{x})$ results from a local physics expressed by the derivatives involved in the differential operator, whereas the regression $u^c(\mathbf{x})$ represents the effect of a hidden local physics. Discovering a differential correction operator could be of major interest in transforming ignorance into knowledge.

By introducing the reference solution $u(\mathbf{x})$ given by Equation (4) into the physics-based model, and taking into account the model linearity and the problem Definition (1), the expression of the residual $\mathcal{R}(\mathbf{x})$ results

$$\mathcal{R}(\mathbf{x}) = -\mathcal{L}^p(u^p(\mathbf{x})) - \mathcal{L}^p(u^c(\mathbf{x})) + F = -\mathcal{L}^p(u^c(\mathbf{x})). \tag{5}$$

Now, it is assumed that a correction $\mathcal{L}^c(\cdot)$ of the physics-based model $\mathcal{L}^p(\cdot)$ must be added, in such a way that

$$\mathcal{L}^p(u(\mathbf{x})) + \mathcal{L}^c(u(\mathbf{x})) = F, \tag{6}$$

that, taking into account Equations (4) and (5), results in

$$\mathcal{L}^c(u(\mathbf{x})) = \mathcal{R}(\mathbf{x}), \tag{7}$$

which will be employed for discovering the linear or nonlinear differential correction operator.

Unlike available techniques published in the literature, the aim of this work is to identify the mathematical expression in the correction operator involved in the mapping of the input forcing function towards the associated output. For example, the input function can be mapped towards the output function through two nonlinear operators multiplying each other [1,2]. However, this method does not derive an explicit expression of the identified operator.

Other methods in the literature aim to approximate the differential operator as a weighted sum of physical functions [3–5], while enforcing sparsity through regularization, offering physically explainable results under the condition of including in the dictionary the right functions. Moreover, classification techniques can be used to create a classifier, which selects the best model to fit the measured data based on optimal trees, that performs correctly as soon as the searched model is involved in the search dictionary [6]. In an effort to enhance the predictability of the outputs as a function of the input variables variations, different architectures can be proposed while considering any information coming from existing physical laws [7]. However, all the proposed architectures aim to create a black box form of the response without actually giving an explainable representation of the physical model.

Differential operator discovery through a machine learning algorithm was recently introduced in [8], where the neural network learns the approximation of an operator while also generating a differential representation of the operator. The proposed strategy aims at approximating at the same time the data points and the residual of the discovered operator while applying regularization. The results prove the ability of the network to identify the operator from a given library of differential operators after several hundreds of evaluations of the solution. Differential equation discovery is also the subject of other works [9], where Bayesian neural networks are used to identify the operator involved in the represented physical problem.

Most of the aforementioned techniques aim to approximate the data response as a whole and not the discrepancy between the results and the physical model. In fact, keeping the physics-based model while enriching it enables explaining and certifying the resulting final hybrid model [10]. In this sense, universal ordinary differential equations aim to complement the physical operator with a neural one; however, they fail to give a physical expression of the generated operator [11].

The present work aims to complement the physical operator with an explainable enrichment (or correction). Moreover, the use of the coordinate separation allows for fast identification of the operators without having to rely on a dictionary, allowing the use of the total number of possible combinations of operators in each dimension. The separated representation allows for fast computation and identification of the solution without requiring numerous iterations of the full-order approximation. Moreover, the proposed strategy accepts a tensor formulation that, as proved in many of our former works, alleviates the curse of dimensionality. The enrichment operator is identified from the physics-based model solution and its residual, both expressed in a separated representation.

The next section proposes a methodology for obtaining the residual separated representation, a sort of sparse Singular Value Decomposition (SVD) [12–14], which circumvents the dimensional limitations of the SVD and extends it beyond the classical High-Order SVD (HOSVD) [15–17]. Later, Sections 3 and 4 will address the construction of the correction operator along with several examples, leveraging the separated representation just discussed. The last section ends the manuscript with some conclusions.

2. Separated Representation: Beyond Usual SVD and HOSVD

The discovery procedures introduced in the next sections are enormously facilitated by the fact of employing separated representations of both the problem solution $u(x,y)$ and the residual $\mathcal{R}(x,y)$, involved in the correction Problem (7). Both separated representations read

$$u(x,y) \approx \sum_{i=1}^{N} \mathcal{X}_i(x)\mathcal{Y}_i(y), \tag{8}$$

and

$$\mathcal{R}(x,y) \approx \sum_{i=1}^{M} \mathcal{G}_i(x)\mathcal{H}_i(y). \tag{9}$$

When operating in a separable domain $\Omega = \Omega^x \times \Omega^y$, a rectangular domain in the 2D case here considered, a grid could be considered equipping Ω, consisting of $K \times L$ nodes $\mathbf{x}_{kl}$, $k = 1,\ldots,K$; $l = 1,\ldots,L$, with coordinates $\mathbf{x}_{kl} = (x_k, y_l)$. The functions $\mathcal{X}_i$ and $\mathcal{Y}_i$ consist of the separated representation of the measured (or real) quantities of interest, while $\mathcal{G}_i$ and $\mathcal{H}_i$ are the separated representation of the residual $\mathcal{R}$. With $\Omega^x = [0, A]$ and $\Omega^y = [0, B]$, the nodal coordinates become

$$\begin{cases} x_k = (k-1)\frac{A}{K-1} & k = 1,\ldots,K, \\ y_l = (l-1)\frac{B}{L-1} & l = 1,\ldots,L. \end{cases} \tag{10}$$

Now, any field ϕ defined in Ω, for instance in our case $u(x,y)$ and $\mathcal{R}(x,y)$, can be presented by a $K \times L$ matrix $\mathbf{Q}$, with the component $\mathbf{Q}_{kl} = \phi(x_k, y_l)$.

Now, the singular value decomposition –SVD– of matrix $\mathbf{Q}$ reads [18]:

$$\mathbf{Q} = \mathbf{U}\mathbf{D}\mathbf{V}^T, \tag{11}$$

with $\mathbf{U}$ and $\mathbf{V}$ defining the orthonormal column and row modes, and $\mathbf{D}$ a diagonal matrix containing the eigenvalues. Vectors in $\mathbf{U}$ and $\mathbf{V}$ represent the y- and x-functions of the eigen-decomposition, whose significance scales with the value of the associated eigenvalue in the diagonal of $\mathbf{D}$.

If we assume the vectors are the discrete nodal representation of the associated continuous functions, the SVD applied to both, the matrix form of $u(x,y)$ and $\mathcal{R}(x,y)$, leads to the Decompositions (8) and (9), respectively.

This procedure works well; however, it is subjected to a quite limiting constraint, the separability of Ω. In many cases, Ω, having a complex shape, equipping it with a regular grid is not possible anymore. An alternative procedure for succeeding with the separation is then needed.

One possibility that follows from our recent work combining proper generalized decomposition—PGD— [19] and physics-informed neural network [20]—PINN—into the so-called PINN-PGD method [21], which consists of assuming the separation of the field under consideration, $u(x,y)$ or $\mathcal{R}(x,y)$ in our case. Considering the N-terms separated representation of $u(x,y)$ given by Equation (8)

$$u^N(x,y) = \sum_{i=1}^{N} \mathcal{X}_i(x)\mathcal{Y}_i(y), \tag{12}$$

the different involved functions $\mathcal{X}_i(x)$ and $\mathcal{Y}_i$, $i = 1,\ldots,N$, are approximated by the $2N$ corresponding neural networks:

$$\begin{cases} \mathcal{X}_i = \mathcal{NN}_i^x(x), \\ \mathcal{Y}_i = \mathcal{NN}_i^y(y), \end{cases} \quad i = 1,\ldots,N, \tag{13}$$

$\mathcal{NN}$ being a dense neural network. All of the neural networks are trained from the loss given by

$$\texttt{Loss} = \sum_{k=1}^{P} \left(\sum_{i=1}^{N} \mathcal{NN}_i^x(x_k)\mathcal{NN}_i^y(y_k) - u(x_k, y_k) \right)^2, \tag{14}$$

where (x_k, y_k) are the coordinates of the collocation points $\mathbf{x}_k$, $k = 1, \ldots, P$, distributed to cover in the best possible way the domain Ω.

As proposed in [21], a greedy strategy is considered for computing the different modes, $\mathcal{X}_i(x)$ and $\mathcal{Y}_i(y)$, one after another. The same rationale applies to decomposing the residual $\mathcal{R}(x, y)$, leading to Expression (9).

One important question concerns the advantages of performing such a coordinates separation instead of looking for a regression, $\hat{u}(x, y)$ and $\hat{\mathcal{R}}(x, y)$, in the whole domain Ω. For example, a Neural Network, among other regressors, could accomplish such an approximation while offering the possibility of computing the different derivatives from its automatic differentiation capabilities. In this work, we select the neural networks as regressors for coordinate separation since they are commonly used as global approximators of nonlinear functions. The main advantages of performing the coordinates separation are:

- The hierarchical approximation benefits, where the different scales of the coordinate correlations are captured gradually. Thus, one could truncate and identify the operators representing the most relevant contributions.
- When operating with a reduced amount of data, the separated representation allows for the extraction of the first (most relevant) components of the decomposition, whereas a monolithic approach exhibits less accuracy for the same amount of data.
- Separated representations facilitate the use of tensor decompositions, especially appealing when addressing highly multidimensional partial differential equations [22].

3. Identification Based on a Full Operator Representation

The operator's identification is first presented in the linear case, then extended to the nonlinear one, before considering a series of numerical tests to prove the proposal performances.

3.1. Linear Operator Discovery

Now that the PINN-PGD method succeeded with the separated decomposition of $u(x, y)$ and $\mathcal{R}(x, y)$, the so-called correction problem

$$\mathcal{L}^c(u(\mathbf{x})) = \mathcal{R}(\mathbf{x}), \tag{15}$$

reads

$$\mathcal{L}^c\left(\sum_{i=1}^{N} \mathcal{X}_i(x)\mathcal{Y}_i(y) \right) = \sum_{j=1}^{M} \mathcal{G}_j(x)\mathcal{H}_j(y), \tag{16}$$

where it is assumed that both $u(x, y)$ and $\mathcal{R}(x, y)$ are approximated by their N and M-terms separated representations, respectively. We recall that $\mathcal{G}_j(x)$ and $\mathcal{H}_j(y)$ are the domain decomposition of the residual through PINN-PGD.

Now, we assume the correction differential operator is expressed as

$$\mathcal{L}^c \approx \sum_{k=0}^{\mathtt{D}} \sum_{l=0}^{\mathtt{D}} c_{kl} \frac{\partial^{k+l}}{\partial x^k \partial y^l}, \tag{17}$$

where, for the sake of simplicity, we assumed the same derivation order (D) applying on both coordinates.

Now, the fact of having performed the separated representation of $u(x,y)$ represents a major benefit, because of the fact that

$$\frac{\partial^{k+l}}{\partial x^k \partial y^l}(\mathcal{X}_i(x)\mathcal{Y}_i(y)) = \frac{\partial^k}{\partial x^k}\mathcal{X}_i(x)\frac{\partial^l}{\partial y^l}\mathcal{Y}_i(y) = \mathcal{D}_i^{k,x}(x)\,\mathcal{D}_i^{l,y}(y), \tag{18}$$

that allows for the expression of the correction problem as

$$\sum_{k=0}^{\mathtt{D}}\sum_{l=0}^{\mathtt{D}}\sum_{i=1}^{N} c_{kl}\,\mathcal{D}_i^{k,x}(x)\,\mathcal{D}_i^{l,y}(y) = \sum_{j=1}^{M}\mathcal{G}_j(x)\mathcal{H}_j(y). \tag{19}$$

Now, sparsity is introduced, promoting the simplest operator, that is, the one containing the minimum number of non-zero coefficients c_{kl}.

One possibility consists of minimizing the residual in a least-squares sense, considering a set of points $\mathbf{x}_p \in \Omega$, $p = 1, \ldots, P$, with coordinates (x_p, y_p)

$$c_{kl} = \mathtt{argmin}_{c_{kl}^*}\left\{\sum_{p=1}^{P}\left(\sum_{k=0}^{\mathtt{D}}\sum_{l=0}^{\mathtt{D}}\sum_{i=1}^{N} c_{kl}^*\mathcal{D}_i^{k,x}(x_p)\mathcal{D}_i^{l,y}(y_p) - \sum_{j=1}^{M}\mathcal{G}_j(x_p)\mathcal{H}_j(y_p)\right)^2 + \lambda\sum_{k=0}^{\mathtt{D}}\sum_{l=0}^{\mathtt{D}}|c_{kl}^*|\right\}, \tag{20}$$

where the last term, involving the $L1$-norm, regularizes the problem and promotes sparsity.

When considering the same number of collocation points $\mathbf{x}_p$ than the number of coefficients c_{kl}, that is $P = \mathtt{D}^2$, the constraints

$$\sum_{k=0}^{\mathtt{D}}\sum_{l=0}^{\mathtt{D}}\sum_{i=1}^{N} c_{kl}^*\mathcal{D}_i^{k,x}(x_p)\mathcal{D}_i^{l,y}(y_p) = \sum_{j=1}^{M}\mathcal{G}_j(x_p)\mathcal{H}_j(y_p), \quad p = 1, \ldots, P, \tag{21}$$

can be expressed using the linear algebraic system

$$\mathbf{M}\mathbf{C}^* = \mathbf{W}, \tag{22}$$

where vector $\mathbf{C}$ contains the different coefficients c_{kl}. Thus, the correction operator construction can be formulated from

$$c_{kl} = \mathtt{argmin}_{c_{kl}^*}\sum_{k=0}^{\mathtt{D}}\sum_{l=0}^{\mathtt{D}}|c_{kl}|^*, \tag{23}$$

which is subjected to the linear constraint

$$\mathbf{M}\mathbf{C}^* = \mathbf{W}. \tag{24}$$

This optimization problem defined in Equations (23) and (24) can therefore be solved by using any linear programming strategy.

3.2. Nonlinear Operator Discovery

The case of considering nonlinear operators follows a quite similar rationale; however, in that case, the correction differential operator includes the different derivatives products. In the case of considering only 2-terms products, the correction reads:

$$\sum_{k=0}^{\mathtt{D}}\sum_{l=0}^{\mathtt{D}}\sum_{m>k}^{\mathtt{D}}\sum_{n>l}^{\mathtt{D}}\sum_{i=1}^{N} c_{klmn}\,\mathcal{D}_i^{k,x}(x)\,\mathcal{D}_i^{m,x}(x)\,\mathcal{D}_i^{l,y}(y)\,\mathcal{D}_i^{n,y}(y), \tag{25}$$

which, after adding to the linear contribution, the procedure described in the previous section can be applied again.

However, this route becomes difficult to implement because of the number of terms exponentially growing. Tensor formats discussed in Section 4 are expected, thus alleviating this issue.

3.3. Numerical Results

In what follows, the reference problems considered will consist of problems extracted from the general heat equation [23,24], expressed as

$$(\alpha_l + \alpha_{nl}u)\frac{\partial^2 u}{\partial x^2} + (\beta_l + \beta_{nl}u)\frac{\partial^2 u}{\partial y^2} + (\gamma_l + \gamma_{nl}u)\frac{\partial^2 u}{\partial x \partial y} + \mu u + \varphi u^2 = \psi, \tag{26}$$

defined in a domain Ω with prescribed temperature on the domain boundary $u|_{\partial\Omega} = y$.

The physics-based model considered in the different cases discussed later reads

$$\alpha_l \frac{\partial^2 u^p}{\partial x^2} + \beta_l \frac{\partial^2 u^p}{\partial y^2} = \psi, \tag{27}$$

with $\alpha_l = \beta_l = 1$ and $\psi = 1$.

3.3.1. Linear Problem: Discovering Anisotropy

In this section, the reference solution is assumed given by

$$\alpha_l \frac{\partial^2 u}{\partial x^2} + \beta_l \frac{\partial^2 u}{\partial y^2} + \gamma_l \frac{\partial^2 u}{\partial x \partial y} = \psi, \tag{28}$$

with $\alpha_l = \beta_l = \gamma_l = 1$ and $\psi = 1$.

The computational domain consists of a 2D rectangle $[-0.5, 0.5] \times [-0.25, 0.25]$. The solution of Equation (28), obtained from a fine enough finite element approximation, is illustrated in Figure 1a. This solution illustrates the linear variation of the thermal fields from the top boundary condition to the bottom one. The discrepancy $u^c(\mathbf{x})$, difference between the physics-based problem solution $u^p(\mathbf{x})$ and the reference one $u(\mathbf{x})$ is illustrated in Figure 1b. This discrepancy illustrates the effect of the term $\frac{\partial^2 u}{\partial x \partial y}$ appearing in the differential equation of $u(\mathbf{x})$, but absent from the equation of $u^p(\mathbf{x})$.

Both the reference $u(\mathbf{x})$ and the residual $\mathcal{R}(\mathbf{x}) = -\mathcal{L}^p(u^c(\mathbf{x}))$ (5) were separated and expressed using a 10-modes decomposition.

By considering the operators given by $\mathtt{D} = 1$ in Equation (17), one expects obtaining $c_{11} = 1$ with all the remaining coefficients vanishing. The regularized identification was performed with $\lambda = 10^{-8}$ (with 10^{-5} being the order of magnitude of the discrepancy), by using the Adam gradient descent algorithm [25,26], and 40 data points randomly distributed into Ω. The identified parameters were: $c_{00} = 0.3$, $c_{10} = 0.2$, $c_{01} = 10^{-5}$, and $c_{11} = 0.9$, exhibiting a quite good identification performance.

It is important to note that by removing those associated with the smallest identified coefficients from the operator's dictionary and applying again the identification procedure, the accuracy increases, to obtain $c_{11} \approx 1$.

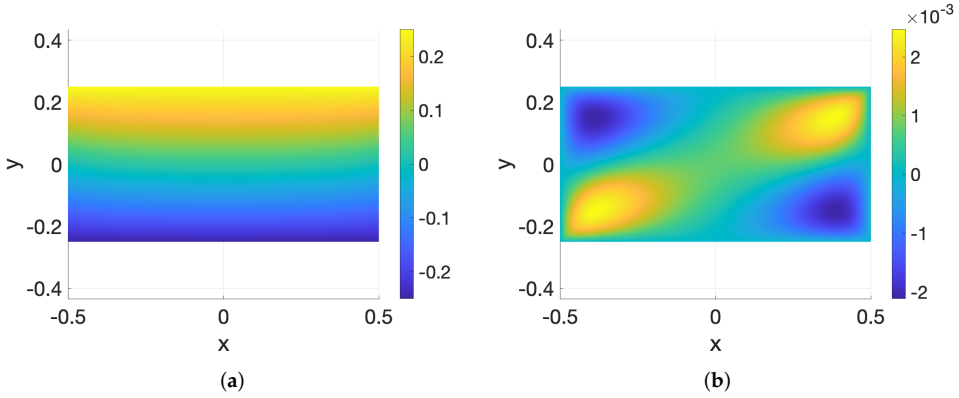


Figure 1. Solution and discrepancy of the problem defined in Equation (28). (**a**) Reference solution $u(\mathbf{x})$ of Equation (28). (**b**) Difference between the reference solution $u(\mathbf{x})$ of the solution of the physics-based model $u^p(\mathbf{x})$, $u^c(\mathbf{x}) = u(\mathbf{x}) - u^p(\mathbf{x})$.

3.3.2. Linear Problem: Discovering Reaction Term

From now on, we consider an elliptical domain Ω, centered at the origin $(0,0)$ with the principal axes aligned with the coordinate system, of lengths $A = 1$ and $B = 0.5$, The solution of Problem (27), obtained from a fine enough finite element discretization, is depicted in Figure 2, for the prescribed boundary condition $u|_{\partial\Omega} = y$. Figure 2 the solution of the differential equation with the absence of the reaction term.

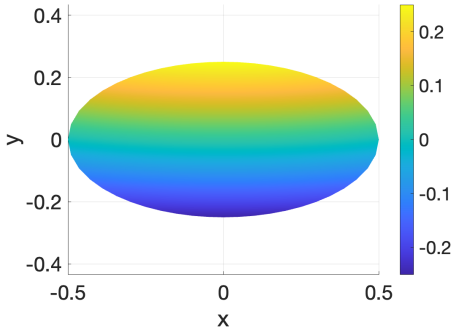


Figure 2. Solution of the physics-based problem $u^p(\mathbf{x})$.

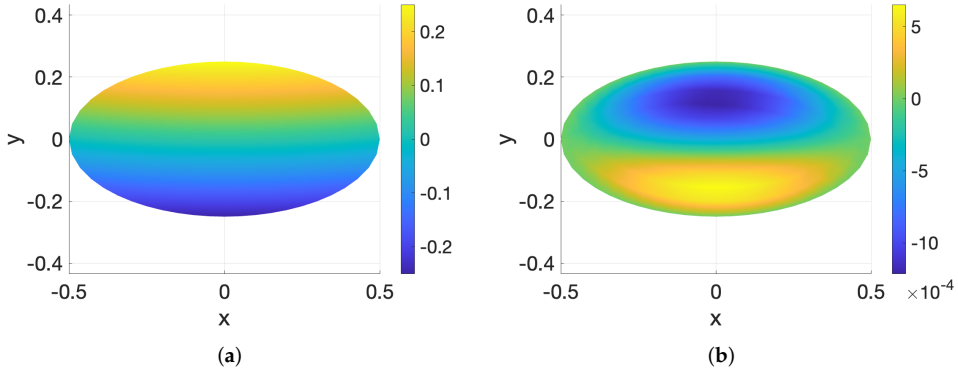
In the present case, the reference solution is assumed to be given by

$$\alpha_l \frac{\partial^2 u}{\partial x^2} + \beta_l \frac{\partial^2 u}{\partial y^2} + \mu u = \psi, \tag{29}$$

with $\alpha_l = \beta_l = 1$, $\mu = 1$ and $\psi = 1$. The solution, obtained from a fine enough finite element approximation, is depicted in Figure 3, along with the discrepancy $u^c(\mathbf{x})$. The discrepancy shows the difference between the solution $u(\mathbf{x})$ and $u^p(\mathbf{x})$, which consists of the contribution of the reaction term to $u(\mathbf{x})$.

By considering again the operators given by $\mathtt{D} = 1$ in Equation (17), one expects obtaining $c_{00} = 1$ with all the remaining coefficients vanishing.

The regularized identification was performed again with $\lambda = 10^{-8}$ and by using the Adam gradient descent algorithm and 40 data points randomly distributed into Ω. The identified parameters were: $c_{00} = 0.62$, $c_{10} = 5.5 \times 10^{-3}$, $c_{01} = 2.9 \times 10^{-4}$ and $c_{11} = 2.3 \times 10^{-3}$, exhibiting again a very good performance.



(**a**) (**b**)

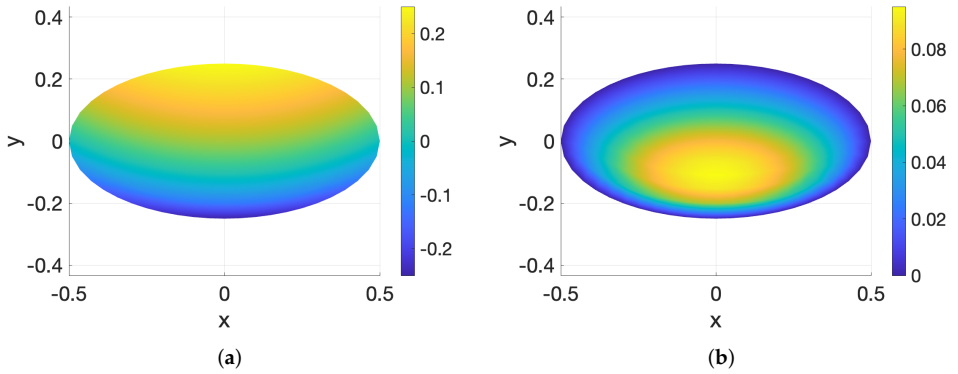
Figure 3. Reference solution $u(\mathbf{x})$ and discrepancy $u^c(\mathbf{x})$ related to Problem (29). (**a**) Reference solution $u(\mathbf{x})$. (**b**) Discrepancy $u^c = u(\mathbf{x}) - u^p(\mathbf{x})$.

3.3.3. Discovering Nonlinear Conductivity

The reference solution is assumed to be given by

$$(\alpha_l + \alpha_{nl}u)\frac{\partial^2 u}{\partial x^2} + (\beta_l + \beta_{nl}u)\frac{\partial^2 u}{\partial y^2} = \psi, \tag{30}$$

with $\alpha_l = \beta_l = 1$, $\alpha_{nl} = \beta_{nl} = 1$ and $\psi = 1$, solved again by using a fine enough finite element discretization. The solution is depicted in Figure 4, along with $u^c(\mathbf{x}) = u(\mathbf{x}) - u^p(\mathbf{x})$. The non-linearity effect is highlighted in the discrepancy, Figure 4, which has an amplitude of about 40% of the solution. This discrepancy has a higher relative amplitude when compared to the linear examples previously depicted.



(**a**) (**b**)

Figure 4. Reference solution $u(\mathbf{x})$ and discrepancy $u^c(\mathbf{x})$ related to the problem (30). (**a**) Reference solution $u(\mathbf{x})$. (**b**) $u^c(\mathbf{x}) = u(\mathbf{x}) - u^p(\mathbf{x})$.

By considering the linear part $\mathtt{D} = 1$ and the nonlinear contribution in Equation (25) related to coefficients c_{0020} and c_{0002} affecting $u\frac{\partial^2 u}{\partial x^2}$ and $u\frac{\partial^2 u}{\partial y^2}$, respectively, these two last coefficients are expected to be close to one, with all the other coefficients involved in the linear operator vanishing, being already represented by the physics-based solution.

The regularized identification was performed with $\lambda = 10^{-8}$ and by using again the Adam gradient descent algorithm, with 40 data points randomly distributed into Ω. Again, a good performance is noticed with the linear contributions coefficients vanishing and the two non-zero taking the values $c_{0020} = 0.88$ and $c_{0002} = 0.64$. The accuracy of the weights is slightly reduced in this example, which can be explained by the errors involved in the approximation forms of the measures, as well as the random selection of a few points for the fitting. Nevertheless, the algorithm discovered the exact missing terms from the

PDE formulation. The accuracy can be improved further by considering only these two operators (associated with the two largest coefficients) in the identification procedure.

3.3.4. Discovering Nonlinear Reaction Term

The reference solution is given by

$$\alpha_l \frac{\partial^2 u}{\partial x^2} + \beta_l \frac{\partial^2 u}{\partial y^2} + \varphi u^2 = \psi, \tag{31}$$

with $\alpha_l = \beta_l = 1$, $\varphi = 1$ and $\psi = 1$, solved again by using the finite element method, with the solution depicted in Figure 5 along with the discrepancy $u^c(\mathbf{x})$. The effect of the nonlinear reaction term used in this example is limited when compared to the actual solution amplitude.

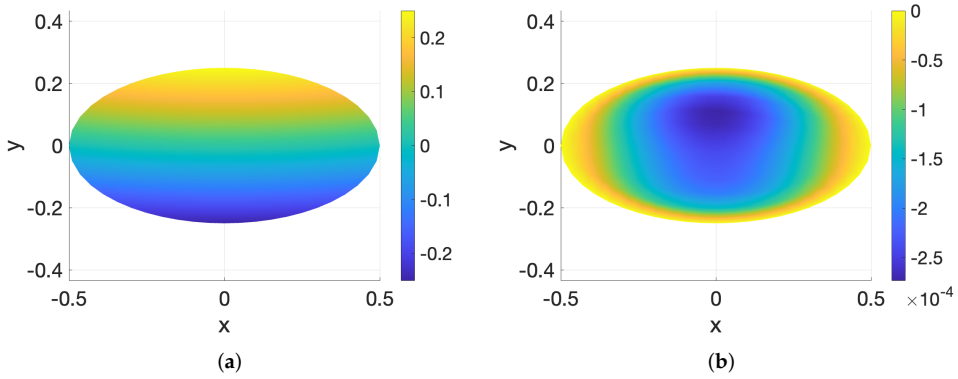


Figure 5. Reference solution $u(\mathbf{x})$ and discrepancy $u^c(\mathbf{x})$ related to the Problem (31). (**a**) Reference solution $u(\mathbf{x})$. (**b**) $u^c(\mathbf{x}) = u(\mathbf{x}) - u^p(\mathbf{x})$.

Now, with respect to the previous section, a new term is added in the nonlinear contribution c_{0000} that is expected to be close to one, with all the others vanishing.

The regularized identification was performed with $\lambda = 10^{-8}$ and by using again the Adam gradient descent algorithm, with 40 data points randomly distributed into Ω. Again, all the coefficients were almost zero, except $c_{0000} = 0.95$.

3.3.5. Discovering Multiple Nonlinear Terms

In this section, the reference solution is given by:

$$(\alpha_l + \alpha_{nl} u) \frac{\partial^2 u}{\partial x^2} + (\beta_l + \beta_{nl} u) \frac{\partial^2 u}{\partial y^2} + \varphi u^2 = \psi, \tag{32}$$

with $\alpha_l = \beta_l = \alpha_{nl} = \beta_{nl} = 1$, $\varphi = 1$ and $\psi = 1$, solved again by using the finite element method, with the solution depicted in Figure 6 along with the discrepancy $u^c(\mathbf{x})$ with respect to the linear solution.

In this example, the solutions includes multiple nonlinear contributions with c_{0000}, c_{0020} and c_{0002} expected to be close to one, with all the other terms vanishing. The regularized identification was performed with $\lambda = 10^{-8}$ and by using again the Adam gradient descent algorithm, with 40 data points randomly selected into Ω. Again, all the coefficients were almost zero, except $c_{0020} = 0.892$, $c_{0002} = 1.102$ and $c_{0000} = 1.012$.

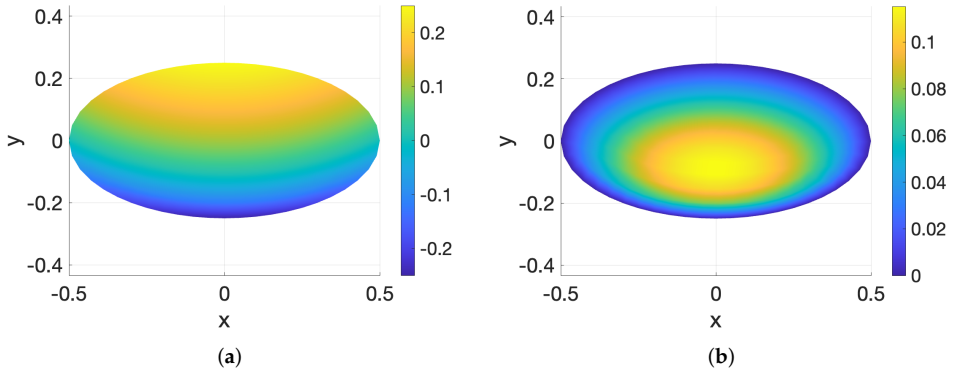


Figure 6. Reference solution $u(\mathbf{x})$ and discrepancy $u^c(\mathbf{x})$ related to the Problem (32). **(a)** Reference solution $u(\mathbf{x})$. **(b)** $u^c(\mathbf{x}) = u(\mathbf{x}) - u^p(\mathbf{x})$.

3.4. Comparing Separated Decomposition vs. Monolithic Approximations

In order to justify the advantage of performing a separation of $u(\mathbf{x})$ and $\mathcal{R}(\mathbf{x})$ with respect to non-separated approximations of both functions based on the use of neural networks, we consider the problem addressed in Section 3.3.3, while considering the neural network-based approximations:

$$\begin{cases} \hat{u} = \mathcal{NN}^u(x, y), \\ \hat{u}^c = \mathcal{NN}^c(x, y). \end{cases} \tag{33}$$

The obtained approximations $\hat{u}(\mathbf{x})$ and $\hat{u}^c(\mathbf{x})$ are depicted in Figure 7 along with the associated approximation error.

Figure 7b,d reveal higher approximation errors than the ones obtained by using separated approximations.

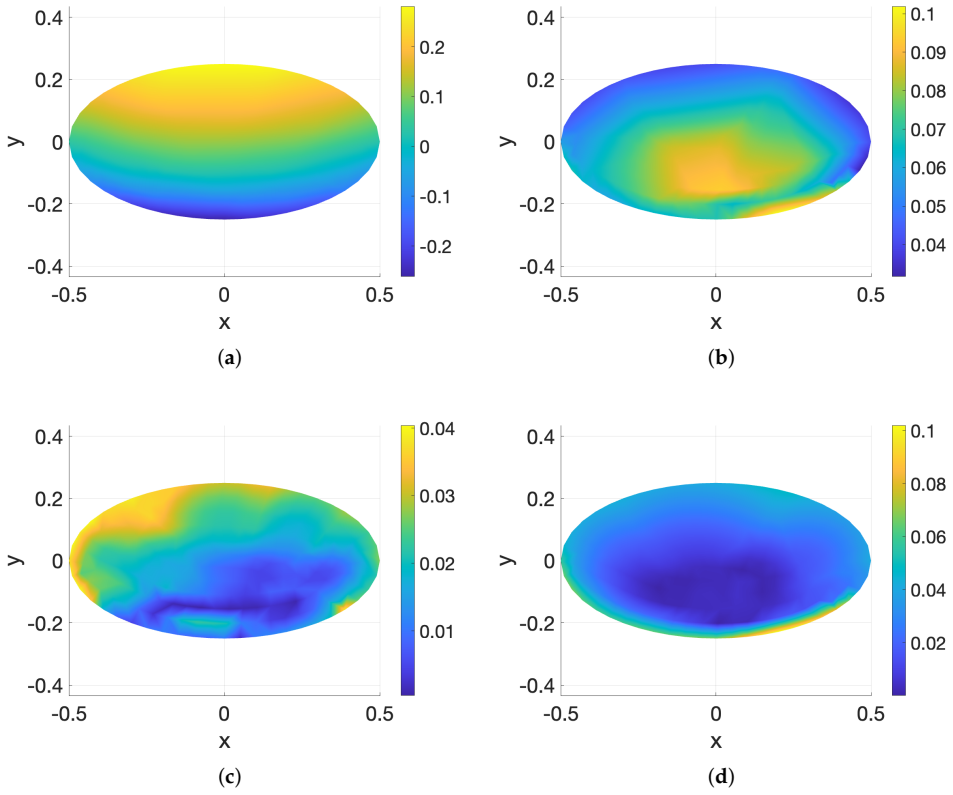


Figure 7. Non-separated approximation of the reference solution and the discrepancy. **(a)** $\hat{u}(\mathbf{x})$. **(b)** $\hat{u}^c(\mathbf{x})$. **(c)** $\hat{u}(\mathbf{x}) - u(\mathbf{x})$. **(d)** $\hat{u}^c(\mathbf{x}) - u^c(\mathbf{x})$.

By using the same identification procedure as in Section 3.3.3 (same operators, regularization, data-points, and optimizer) the two non-zero coefficients were obtained with larger errors: $c_{0020} = 2.44$ and $c_{0002} = 1.74$. This result highlighted the advantage of using domain decomposition over monolithic approximations from a precision point of view. Another obvious advantage is the complexity scaling linearly with respect to the dimensionality of the problem when using domain decomposition, as compared to exponential scaling when monolithic approximations are employed.

4. A Tensor Formulation of the Identification Problem

The previous section considered the full operator representation:

$$\mathcal{L}^c \approx \sum_{k=0}^{\mathrm{D}} \sum_{l=0}^{\mathrm{D}} c_{kl} \frac{\partial^{k+l}}{\partial x^k \partial y^l}, \tag{34}$$

however, very few coefficients c_{kl} are expected since they are non-zero, a fact that was taken into account by using an appropriate solution procedure ($L1$-regularization).

In this section, more than considering all the possible terms, to finally select the non-zero ones, we propose extracting those that are non-zero directly.

Such a procedure becomes especially appealing for addressing multidimensional partial differential equations –PDE–, as those encountered in the kinetic theory description of macromolecules from the use of the so-called Fokker–Planck formalism [22]. This approach is especially suitable for multidimensional problems, as the complexity scales linearly with the separated dimensions, unlike the problems where the decomposition is not applied, and hence the problem complexity scales exponentially as a function of the problem's dimensionality [18].

4.1. Linear Case

We assume the operator applied on a generic separated function $\mathcal{X}(x)\mathcal{Y}(y)$:

$$\mathcal{L}^c(\mathcal{X}(x)\mathcal{Y}(y)) \approx \sum_{j=0}^{\mathrm{J}} \mathbf{a}_j^T \mathbf{D}^x \mathbf{D}^{y^T} \mathbf{b}_j, \tag{35}$$

where $\mathbf{D}^x$ is given by

$$\mathbf{D}^x = \begin{Bmatrix} \mathcal{X}(x) \\ \frac{\partial \mathcal{X}(x)}{\partial x} \\ \frac{\partial^2 \mathcal{X}(x)}{\partial x^2} \\ \vdots \\ \frac{\partial^{\mathrm{p}} \mathcal{X}(x)}{\partial x^{\mathrm{p}}} \end{Bmatrix}, \tag{36}$$

and similarly for $\mathbf{D}^y$

$$\mathbf{D}^y = \begin{Bmatrix} \mathcal{Y}(y) \\ \frac{\partial \mathcal{Y}(y)}{\partial y} \\ \frac{\partial^2 \mathcal{Y}(y)}{\partial y^2} \\ \vdots \\ \frac{\partial^{\mathrm{p}} \mathcal{Y}(y)}{\partial y^{\mathrm{p}}} \end{Bmatrix}. \tag{37}$$

Now, applied on the N-terms separated representation of $u(x, y)$ given by Equation (8), results in:

$$\mathcal{L}^c \left(\sum_{i=1}^{N} \mathcal{X}_i(x)\mathcal{Y}_i(y) \right) \approx \sum_{j=0}^{\mathrm{J}} \mathbf{a}_j^T \sum_{i=1}^{N} \left\{ \mathbf{D}_i^x(x) \mathbf{D}_i^{y^T}(y) \right\} \mathbf{b}_j, \tag{38}$$

where the index $\bullet_i$ in vectors $\mathbf{D}^x$ and $\mathbf{D}^y$ indicate that the associated derivatives apply on $\mathcal{X}_i(x)$ and $\mathcal{Y}_i(y)$, respectively.

We can define the matrices

$$\mathbb{D}^{xy}(x,y) = \sum_{i=1}^{N} \left\{ \mathbf{D}_i^x(x) \mathbf{D}_i^{y^T}(y) \right\}, \tag{39}$$

and its transpose

$$\mathbb{D}^{yx}(x,y) = \sum_{i=1}^{N} \left\{ \mathbf{D}_i^y(y) \mathbf{D}_i^{x^T}(x) \right\}, \tag{40}$$

that allows for the writing of two equivalent expressions

$$\mathcal{L}^c \left(\sum_{i=1}^{N} \mathcal{X}_i(x) \mathcal{Y}_i(y) \right) \approx \sum_{j=0}^{\mathtt{J}} \mathbf{a}_j^T \mathbb{D}^{xy}(x,y) \mathbf{b}_j, \tag{41}$$

or

$$\mathcal{L}^c \left(\sum_{i=1}^{N} \mathcal{X}_i(x) \mathcal{Y}_i(y) \right) \approx \sum_{j=0}^{\mathtt{J}} \mathbf{b}_j^T \mathbb{D}^{yx}(x,y) \mathbf{a}_j, \tag{42}$$

specially suitable for calculating coefficients $\mathbf{b}_j$ and $\mathbf{a}_j$, respectively, as described just below.

We assume that at iteration $\mathcal{J} < \mathtt{J}$, the already computed operator is approximated from

$$\mathcal{L}^c_{\mathcal{J}-1} \left(\sum_{i=1}^{N} \mathcal{X}_i(x) \mathcal{Y}_i(y) \right) = \sum_{j=1}^{\mathcal{J}-1} \mathbf{a}_j^T \mathbb{D}^{xy}(x,y) \mathbf{b}_j, \tag{43}$$

with the update

$$\mathcal{L}^c_{\mathcal{J}} \left(\sum_{i=1}^{N} \mathcal{X}_i(x) \mathcal{Y}_i(y) \right) = \mathcal{L}^c_{\mathcal{J}-1} \left(\sum_{i=1}^{N} \mathcal{X}_i(x) \mathcal{Y}_i(y) \right) + \mathbf{a}_{\mathcal{J}}^T \mathbb{D}^{xy}(x,y) \mathbf{b}_{\mathcal{J}}. \tag{44}$$

Thus, the resulting identification equation reads

$$\mathcal{L}^c_{\mathcal{J}-1} \left(\sum_{i=1}^{N} \mathcal{X}_i(x) \mathcal{Y}_i(y) \right) + \mathbf{a}_{\mathcal{J}}^T \mathbb{D}^{xy}(x,y) \mathbf{b}_{\mathcal{J}} = \sum_{m=1}^{M} \mathcal{G}_m(x) \mathcal{H}_m(y), \tag{45}$$

whose residual at interaction $\mathcal{J}$ is defined by

$$f_{\mathcal{J}}(x,y) = \sum_{m=1}^{M} \mathcal{G}_m(x) \mathcal{H}_m(y) - \mathcal{L}^c_{\mathcal{J}-1} \left(\sum_{i=1}^{N} \mathcal{X}_i(x) \mathcal{Y}_i(y) \right), \tag{46}$$

that allows for the rewriting the identification equation as

$$\mathbf{a}_{\mathcal{J}}^T \mathbb{D}^{xy}(x,y) \mathbf{b}_{\mathcal{J}} = f_{\mathcal{J}}(x,y), \tag{47}$$

where both vectors $\mathbf{a}_{\mathcal{J}}$ and $\mathbf{b}_{\mathcal{J}}$ must be computed.

Since the problem is nonlinear, an iteration procedure must be applied. The simplest one, an alternate directions fixed point iteration, calculates $\mathbf{b}_{\mathcal{J}}$ from the just computed $\mathbf{a}_{\mathcal{J}}$, before updating the last from the just computed $\mathbf{b}_{\mathcal{J}}$. The procedure continues until convergence (the fixed point) is reached.

When computing $\mathbf{b}_{\mathcal{J}}$, the previous form of the identification equation is suitable

$$\mathbf{a}_{\mathcal{J}}^T \mathbb{D}^{xy}(x,y) \mathbf{b}_{\mathcal{J}} = f_{\mathcal{J}}(x,y), \tag{48}$$

whereas when calculating $\mathbf{a}_{\mathcal{J}}$ the same equation is rewritten as

$$\mathbf{b}_{\mathcal{J}}^{T}\mathbb{D}^{yx}(x,y)\mathbf{a}_{\mathcal{J}} = f_{\mathcal{J}}(x,y), \tag{49}$$

to facilitate the solution procedure described below.

Both Equations (48) and (49) apply for the available data, at points (x_p, y_p), $p = 1, \ldots, P$. With $\mathbf{c}_{\mathcal{J}}^{T}(x_p, y_p) = \mathbf{a}_{\mathcal{J}}^{T}\mathbb{D}^{xy}(x_p, y_p)$ the results are:

$$\mathbf{c}_{\mathcal{J}}^{T}(x_p, y_p)\mathbf{b}_{\mathcal{J}} = f_{\mathcal{J}}(x_p, y_p), \tag{50}$$

and with $\mathbf{d}_{\mathcal{J}}^{T}(x_p, y_p) = \mathbf{b}_{\mathcal{J}}^{T}\mathbb{D}^{yx}(x_p, y_p)$

$$\mathbf{d}_{\mathcal{J}}^{T}(x_p, y_p)\mathbf{a}_{\mathcal{J}} = f_{\mathcal{J}}(x_p, y_p), \tag{51}$$

from which the following matrix forms are obtained

$$\begin{pmatrix} \mathbf{c}_{\mathcal{J}}^{T}(x_1, y_1) \\ \vdots \\ \mathbf{c}_{\mathcal{J}}^{T}(x_p, y_p) \\ \vdots \\ \mathbf{c}_{\mathcal{J}}^{T}(x_P, y_P) \end{pmatrix} \mathbf{b}_{\mathcal{J}} = \begin{pmatrix} f_{\mathcal{J}}(x_1, y_1) \\ \vdots \\ f_{\mathcal{J}}(x_p, y_p) \\ \vdots \\ f_{\mathcal{J}}(x_P, y_P) \end{pmatrix}, \tag{52}$$

and

$$\begin{pmatrix} \mathbf{d}_{\mathcal{J}}^{T}(x_1, y_1) \\ \vdots \\ \mathbf{d}_{\mathcal{J}}^{T}(x_p, y_p) \\ \vdots \\ \mathbf{d}_{\mathcal{J}}^{T}(x_P, y_P) \end{pmatrix} \mathbf{a}_{\mathcal{J}} = \begin{pmatrix} f_{\mathcal{J}}(x_1, y_1) \\ \vdots \\ f_{\mathcal{J}}(x_p, y_p) \\ \vdots \\ f_{\mathcal{J}}(x_P, y_P) \end{pmatrix}, \tag{53}$$

that can be expressed in a more compact form, from

$$\hat{\mathbf{c}}_{\mathcal{J}}\mathbf{b}_{\mathcal{J}} = \mathbf{f}_{\mathcal{J}}, \tag{54}$$

and

$$\hat{\mathbf{d}}_{\mathcal{J}}\mathbf{a}_{\mathcal{J}} = \mathbf{f}_{\mathcal{J}}, \tag{55}$$

whose least squares formulation reads

$$\hat{\mathbf{c}}_{\mathcal{J}}^{T}\hat{\mathbf{c}}_{\mathcal{J}}\mathbf{b}_{\mathcal{J}} = \hat{\mathbf{c}}_{\mathcal{J}}^{T}\mathbf{f}_{\mathcal{J}} \rightarrow \mathbf{C}_{\mathcal{J}}\mathbf{b}_{\mathcal{J}} = \mathbf{F}_{\mathcal{J}}, \tag{56}$$

and

$$\hat{\mathbf{d}}_{\mathcal{J}}^{T}\hat{\mathbf{d}}_{\mathcal{J}}\mathbf{a}_{\mathcal{J}} = \hat{\mathbf{d}}_{\mathcal{J}}^{T}\mathbf{f}_{\mathcal{J}} \rightarrow \mathbf{D}_{\mathcal{J}}\mathbf{a}_{\mathcal{J}} = \mathbf{F}_{\mathcal{J}}, \tag{57}$$

that must be solved iteratively until reaching the fixed point, before moving to the next step $\mathcal{J} + 1$.

The so-called *ridge* regularized formulation, that tries to prevent overfitting, reads:

$$(\mathbf{C}_{\mathcal{J}} + \lambda\mathbf{I})\mathbf{b}_{\mathcal{J}} = \mathbf{F}_{\mathcal{J}}, \tag{58}$$

and

$$(\mathbf{D}_{\mathcal{J}} + \lambda\mathbf{I})\mathbf{a}_{\mathcal{J}} = \mathbf{F}_{\mathcal{J}}, \tag{59}$$

both associated with the minimization problems

$$\mathbf{b}_{\mathcal{J}} = \texttt{argmmin}_{\mathbf{b}^*_{\mathcal{J}}} \|\mathbf{C}_{\mathcal{J}}\mathbf{b}^*_{\mathcal{J}} - \mathbf{F}_{\mathcal{J}}\|_2^2 + \lambda\|\mathbf{b}^*_{\mathcal{J}}\|_2^2, \tag{60}$$

and

$$\mathbf{a}_{\mathcal{J}} = \texttt{argmmin}_{\mathbf{a}^*_{\mathcal{J}}} \|\mathbf{D}_{\mathcal{J}}\mathbf{a}^*_{\mathcal{J}} - \mathbf{F}_{\mathcal{J}}\|_2^2 + \lambda\|\mathbf{a}^*_{\mathcal{J}}\|_2^2, \tag{61}$$

that can be generalized, combining ridge and Lasso to also promote sparsity, according to

$$\mathbf{b}_{\mathcal{J}} = \texttt{argmmin}_{\mathbf{b}^*_{\mathcal{J}}} \|\mathbf{C}_{\mathcal{J}}\mathbf{b}^*_{\mathcal{J}} - \mathbf{F}_{\mathcal{J}}\|_2^2 + \lambda\Big\{(1-\alpha)\|\mathbf{b}^*_{\mathcal{J}}\|_2^2 + \alpha\|\mathbf{b}^*_{\mathcal{J}}\|_1\Big\}, \tag{62}$$

and

$$\mathbf{a}_{\mathcal{J}} = \texttt{argmmin}_{\mathbf{a}^*_{\mathcal{J}}} \|\mathbf{D}_{\mathcal{J}}\mathbf{a}^*_{\mathcal{J}} - \mathbf{F}_{\mathcal{J}}\|_2^2 + \lambda\Big\{(1-\alpha)\|\mathbf{a}^*_{\mathcal{J}}\|_2^2 + \alpha\|\mathbf{a}^*_{\mathcal{J}}\|_1\Big\}, \tag{63}$$

where α weights the ridge versus the Lasso contributions. The procedure generalizes to the highly-multidimensional case as proved in [27].

4.2. Nonlinear Case

The nonlinear case becomes much more complex; however, when considering nonlinearities induced by the operator's product, the formulation simplifies a bit.

The linear case, applied on $\mathcal{X}(x)\mathcal{Y}(y)$, considered the expression

$$\sum_{j=0}^{\mathrm{J}} \mathbf{a}_j^T \mathbf{D}^x \mathbf{b}_j^T \mathbf{D}^y. \tag{64}$$

When considering nonlinear 2-terms products, it results in

$$\sum_{j=0}^{\mathrm{J}} ({\mathbf{a}_j^1}^T \mathbf{D}^x {\mathbf{b}_j^1}^T \mathbf{D}^y) \cdot ({\mathbf{a}_j^2}^T \mathbf{D}^x {\mathbf{b}_j^2}^T \mathbf{D}^y), \tag{65}$$

and so on for the higher terms products.

Now, the linearization concerns both the alternated directions and also the nonlinear terms.

4.3. Discussion

The tensor formulation just presented is not designed to find the right coefficients affecting the different derivatives composing the searched operator but to identify the different derivatives involved in that search for differential operators. Because of the iteration procedure, the different approximations involved in the separated representations, and the regularizations with repeated and accumulated effects, one expects to capture the significance of the involved derivatives more than computing accurately the coefficient values affecting them.

Thus, as soon as the tensor formulation identifies the missing terms (derivatives), one could consider the full-basis representation addressed in the previous section to calculate the coefficients affecting the different derivatives involved in the enrichment differential operator.

To verify these expectations, the problem addressed in Section 3.3.1 is addressed again, now by first employing the tensor formulation. The procedure was able to identify the significance of the term representing $\frac{\partial^2}{\partial x \partial y}$, which resulted in orders of magnitude higher than the others, proving that the tensor formulation enables an efficient identification. However, the tensor formulation failed to offer the right weight of the discovered term, with a calculated value three orders of magnitude smaller than the reference one. The tensor formulation is, therefore, suitable to detect the missing terms in the model, but it is more convenient to use the approach described in Section 3 to find the corresponding

weights of the missing terms. However, the advantage of the tensor formulation is the ability to use a large number of possible terms without increasing the computation time required for the problem's solution or evaluation.

Concerning the regularization parameters, it was noted that increasing the regularization parameter helps in sparsity and identifying the exact missing terms in the PDE but compromises the identification of the exact weights. Therefore, a suitable approach would be to use a large regularization parameter for terms identification, followed by another fitting, keeping only the identified terms from the PDE while selecting a low regularization parameter.

From a computation time point of view, when using the least squares programming, which is possible for both proposed strategies, the identification is quasi-instant as it only involves inverting an $m \times m$ matrix, where m is the number of operators involved in the approximation. Using a gradient descent algorithm would involve a slightly higher computation time, as it involves an iterative gradient descent process. Nevertheless, the identification time is around 120 s for all the proposed examples. The separated representation approximation computation time depends heavily on the number of modes involved in the approximation. This approximation required between three and five minutes for convergence, for all the examples presented in this manuscript.

5. Conclusions

This paper proposed two different procedures for identifying the differential operators representing the discrepancy between the solution (known at some locations) and the physics-based solution.

First, the physics-based model solution and the discrepancy approximation are expressed in a separate form. Then, a dictionary of differential operators, or its associated tensorial approximation, both regularized, are employed for identifying the most relevant terms expected involved in the discrepancy operator.

The pertinence of the separated representations, as well as the regularized identification procedure, were proved by a series of numerical tests.

Author Contributions: Conceptualization, C.G. and F.C.; methodology, C.G. and F.C.; software, C.G.; validation, C.G. and F.C.; formal analysis, C.G.; investigation, C.G.; resources, F.C.; data curation, C.G.; writing—original draft preparation, C.G.; writing—review and editing, C.G. and F.C.; visualization, C.G.; supervision, F.C.; project administration, F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data generator is contained within the article.

Acknowledgments: Authors acknowledge the support of the ESI Group, RTE and SKF through their research chairs at Arts et Metiers Institute of Technology.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lu, L.; Jin, P.; Pang, G.; Zhang, Z.; Karniadakis, G.E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* **2021**, *3*, 218–229. [CrossRef]
2. Li, W.; Bazant, M.Z.; Zhu, J. Phase-Field DeepONet: Physics-informed deep operator neural network for fast simulations of pattern formation governed by gradient flows of free-energy functionals. *Comput. Methods Appl. Mech. Eng.* **2023**, *416*, 116299. [CrossRef]

3. Brunton, S.L.; Proctor, J.L.; Kutz, J.N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. USA* **2016**, *113*, 3932–3937. [CrossRef] [PubMed]
4. de Silva, B.M.; Higdon, D.M.; Brunton, S.L.; Kutz, J.N. Discovery of Physics From Data: Universal Laws and Discrepancies. *Front. Artif. Intell.* **2020**, *3*, 25. [CrossRef]
5. Egan, K.; Li, W.; Carvalho, R. Automatically discovering ordinary differential equations from data with sparse regression. *Commun. Phys.* **2024**, *7*, 20. [CrossRef]
6. Kapteyn, M.G.; Willcox, K.E. From Physics-Based Models to Predictive Digital Twins via Interpretable Machine Learning. *arXiv* **2020**, arXiv:2004.11356.
7. Iten, R.; Metger, T.; Wilming, H.; del Rio, L.; Renner, R. Discovering Physical Concepts with Neural Networks. *Phys. Rev. Lett.* **2020**, *124*, 010508. [CrossRef]
8. Stephany, R.; Earls, C. PDE-LEARN: Using deep learning to discover partial differential equations from noisy, limited data. *Neural Netw.* **2024**, *174*, 106242. [CrossRef]
9. Hvatov, A.; Titov, R. Towards true discovery of the differential equations. *arXiv* **2024**, arXiv:2308.04901.
10. Ghnatios, C.; Rodriguez, S.; Tomezyk, J.; Dupuis, Y.; Mouterde, J.; Da Silva, J.; Chinesta, F. A hybrid twin based on machine learning enhanced reduced order model for real-time simulation of magnetic bearings. *Adv. Model. Simul. Eng. Sci.* **2024**, *11*, 3. [CrossRef]
11. Rackauckas, C.; Ma, Y.; Martensen, J.; Warner, C.; Zubov, K.; Supekar, R.; Skinner, D.; Ramadhan, A.; Edelman, A. Universal Differential Equations for Scientific Machine Learning. *arXiv* **2021**, arXiv:2001.04385.
12. Stewart, G. On the Early History of the Singular Value Decomposition. *SIAM Rev.* **1993**, *35*, 551–566. [CrossRef]
13. Amsallem, D.; Zahr, M.J.; Farhat, C. Nonlinear model order reduction based on local reduced-order bases. *Int. J. Numer. Methods Eng.* **2012**, *92*, 891–916. [CrossRef]
14. Weiss, J. A Tutorial on the Proper Orthogonal Decomposition. In Proceedings of the 2019 AIAA Aviation Forum, Dallas, TX, USA, 17–21 June 2019.
15. Rövid, A.; Szeidl, L.; Várlaki, P. Data representation in HOSVD-DCT based domain. In Proceedings of the 2013 IEEE 17th International Conference on Intelligent Engineering Systems (INES), San Jose, Costa Rica, 19–21 June 2013; pp. 103–106. [CrossRef]
16. Kolda, T.G.; Bader, B.W. Tensor Decompositions and Applications. *SIAM Rev.* **2009**, *51*, 455–500. [CrossRef]
17. Bader, B.W.; Kolda, T.G. Efficient MATLAB Computations with Sparse and Factored Tensors. *SIAM J. Sci. Comput.* **2008**, *30*, 205–231. [CrossRef]
18. Ghnatios, C.; Abisset, E.; Ammar, A.; Cueto, E.; Duval, J.L.; Chinesta, F. Advanced separated spatial representations for hardly separable domains. *Comput. Methods Appl. Mech. Eng.* **2019**, *354*, 802–819. [CrossRef]
19. Sancarlos, A.; Ghnatios, C.; Duval, J.L.; Zerbib, N.; Cueto, E.; Chinesta, F. Fast computation of multi-parametric electromagnetic fields in synchronous machines by using pgd-based fully separated representations. *Energies* **2021**, *14*, 1454. [CrossRef]
20. Cuomo, S.; Di Cola, V.S.; Giampaolo, F.; Rozza, G.; Raissi, M.; Piccialli, F. Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What's Next. *J. Sci. Comput.* **2022**, *92*, 88. [CrossRef]
21. Ghnatios, C.; Chinesta, F. A Parsimonious Separated Representation Empowering PINN–PGD-Based Solutions for Parametrized Partial Differential Equations. *Mathematics* **2024**, *12*, 2365. [CrossRef]
22. Binetruy, C.; Chinesta, F.; Keunings, R. *Flows in Polymers, Reinforced Polymers and Composites: A Multiscale Approach*; Springerbrief; Springer: Berlin/Heidelberg, Germany, 2015.
23. Myers, G.E. *Analytical Methods in Conduction Heat Transfer*; McGraw-Hill College: New York, NY, USA, 1971.
24. Cotta, R.M.; Knupp, D.C.; Quaresma, J.N.N. Analytical Methods in Heat Transfer. In *Handbook of Thermal Science and Engineering*; Springer International Publishing: Cham, Switzerland, 2018; pp. 61–126.
25. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980.
26. Reddi, S.J.; Kale, S.; Kumar, S. On the Convergence of Adam and Beyond. *arXiv* **2019**, arXiv:1904.09237.
27. Sancarlos, A.; Champaney, V.; Cueto, E.; Chinesta, F. Regularized regressions for parametric models based on separated representations. *Adv. Model. Simul. Eng. Sci.* **2023**, *10*, 4. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.