



Combining physics-based and data-driven techniques for reliable hybrid analysis and modeling using the corrective source term approach

Sindre Stenen Blakseth ^{a,1}, Adil Rasheed ^{b,d,*}, Trond Kvamsdal ^{c,d}, Omer San ^e

^a Department of Physics, Norwegian University of Science and Technology, Norway

^b Department of Engineering Cybernetics, Norwegian University of Science and Technology, Norway

^c Department of Mathematical Sciences, Norwegian University of Science and Technology, Norway

^d Department of Mathematics and Cybernetics, SINTEF Digital, Norway

^e School of Mechanical and Aerospace Engineering, Oklahoma State University, United States of America

ARTICLE INFO

Article history:

Received 31 January 2022

Received in revised form 7 July 2022

Accepted 13 August 2022

Available online 19 August 2022

Keywords:

Deep neural networks

Reliable hybrid analysis and modeling

Physics-based modeling

Data-driven modeling

ABSTRACT

Upcoming technologies like digital twins, autonomous, and artificial intelligent systems involving **safety-critical applications** require accurate, interpretable, computationally efficient, and generalizable models. Unfortunately, the two most commonly used modeling approaches, **physics-based modeling** (PBM) and **data-driven modeling** (DDM) struggle to satisfy all these requirements. In the current work, we demonstrate how a hybrid approach **combining the best of PBM and DDM** can result in models that can outperform both of them. We do so by combining partial differential equations based on first principles describing partially known physics with a black box DDM, in this case, a deep neural network model compensating for the unknown physics. The **novelty of the work** is in the theoretical contribution of presenting a sound **mathematical argument** for why the **approach should work**. The argument is backed by an array of experiments involving a two-dimensional heat diffusion problem with unknown source terms. **The hybrid approach demonstrates the method's superior performance in accuracy and generalizability**. Additionally, it is shown how the DDM part can be interpreted within the hybrid framework to make the overall **approach reliable**. The approach, as we see, will be a door opener for underutilized DDMs in high stake applications.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The development and industrial utilization of digital twins (DTs) is an important trend facilitated by the increased digitalization following Industry 4.0, in addition to recent advances within data processing, computational infrastructure and big data cybernetics. DTs [1] can be defined as virtual representations of physical assets and their applications include real-time prediction, optimization, monitoring, control, and improved decision making. For DTs to be successful in these applications, high-quality modeling techniques are paramount. In particular, DTs require models which are generalizable, trustworthy, self-evolving and computationally efficient while maintaining good accuracy [2]. Unfortunately, it has proven difficult to attain all four of these

modeling characteristics using either of the two traditional modeling paradigms: physics-based modeling (PBM) and data-driven modeling (DDM).

PBM (Fig. 1) generally describes the system to be modeled using a set of governing equations representing known and understood physical phenomena. However, the governing equations generally do **not reflect the complete physics** of the system, as some relevant physics may be unobserved, not understood, or neglected as a **simplifying assumption**. In addition, further loss of physics may result from solving the governing equations using numerical methods with finite precision. As such, PBM generally does not capture the complete physics of the system being modeled. However, they are still considered **trustworthy** because we know exactly which physical phenomena are included in the model and can **bound their numerical errors**. PBM also tends to generalize well, as it is typically not fine-tuned for specific applications. However, solving the governing equations can be **computationally expensive**. Another disadvantage of PBM is that it is generally static in the sense that it **does not automatically get updated** to account for **new scenarios** encountered after model deployment.

* Corresponding author at: Department of Engineering Cybernetics, Norwegian University of Science and Technology, Norway.

E-mail addresses: sindre.blakseth@sintef.no (S.S. Blakseth), adil.rasheed@ntnu.no (A. Rasheed), trond.kvamsdal@ntnu.no (T. Kvamsdal), osan@okstate.edu (O. San).

¹ Currently at the Department of Gas Technology, SINTEF Energy Research.

Nomenclature

BC	Boundary Condition
CoSTA	Corrective Source Term Approach
DDM	Data-Driven Model(ing)
DNN	Deep Neural Network
DT	Digital Twin
FC	Fully Connected
FVM	Finite Volume Method
HAM	Hybrid Analysis and Modeling
MSE	Mean Squared Error
MMS	Method of Manufactured Solutions
NN	Neural Network
PBM	Physics-Based Model(ing)
PDE	Partial Differential Equation
PGNN	Physics-Guided Neural Network
PINN	Physics-Informed Neural Network
ROM	Reduced-Order Model
T	Temperature
T_{ref}	Reference temperature field
$T_p/T_d/T_h$	Temperature field predicted by PBM/DDM/CoSTA
T_e, T_w, T_n, T_s	Boundary temperatures
$\hat{\sigma}/\hat{\sigma}_{\text{NN}}$	Reference/DNN-generated corrective source term
$\hat{\sigma}_p/\hat{\sigma}_k$	Source term correcting error in modeling of P/k
$\mathcal{N}_\Omega/\mathcal{N}_{\partial\Omega}$	Operators defining general PDE
f/g	Right-hand-side functions defining general PDE
u	True solution of general PDE
k	Thermal conductivity
P	Internal heat generation rate
c_v	Specific heat capacity at constant volume
ρ	Density
α	General system parameter
$x/y/t$	Spatial and temporal coordinates
$E_p/E_d/E_h$	Relative ℓ_2 -errors of PBM/DDM/CoSTA
ϵ_p/ϵ_k	Error in modeling of P/k
ξ	Convenience function (see Eq. (26))
$\sim/\hat{\sim}$	Approximation/Corrected approximation
$j,i/l^n$	Grid cell indices/Time level
$\text{DNN}_\sigma/\text{DNN}_T$	DNN predicting $\hat{\sigma}/T$
N_t	Maximum time level

DDM (Fig. 2) exhibits the opposite traits of PBM, as DDM is generally inexpensive to run and can be continuously updated using new data even after deployment. Furthermore, the fact that observations may include neglected/unknown physics means that DDM, if tuned and trained perfectly, can reflect the complete physics. However, under realistic (imperfect) conditions, DDM will be biased towards the data samples on which the model is trained. This limits the models' generalizability, especially to extrapolation scenarios. Additionally, even with recent research focused on increasing the explainability of DDM, it can be challenging to establish precisely what physics are modeled by a DDM

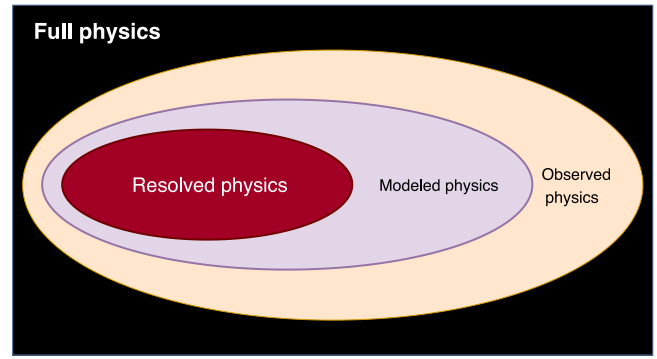


Fig. 1. PBM: black part corresponds to unknown/unmodeled physics, orange ellipse corresponds to observed physics, purple ellipse corresponds to actually modeled physics while red ellipse signifies the actual physics solved for.

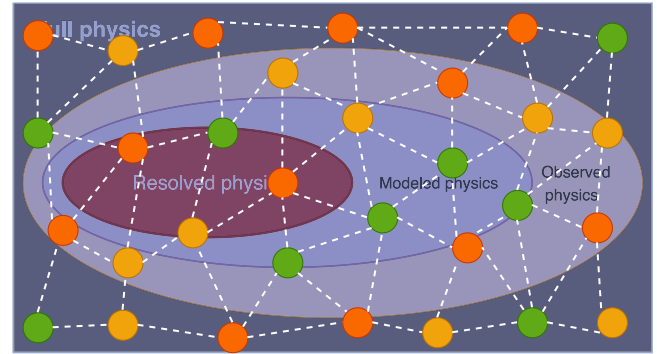


Fig. 2. DDM: It is assumed that since data is a manifestation of both known and unknown physics, models trained on the data will implicitly capture full physics.

approach. This black-box-like nature greatly hurts the trustworthiness of DDM and is the primary barrier keeping DDM from entering high-stakes and safety-critical applications.

From an analysis of the strong and weak points of PBM and DDM as described above, it is clear that neither modeling approach is ideal for use in DTs. However, we see that all four modeling characteristics identified by San et al. [2] can be attained by combining PBM and DDM in a way that retains their strengths while eliminating their weaknesses. This is exactly the philosophy behind the emerging Hybrid Analysis and Modeling (HAM) paradigm. As shown in Fig. 3, HAM utilizes PBM to the maximum extent possible, and only compensates for the unmodeled/unknown physics using DDM. Recent works have explored many interesting approaches to HAM, most of which falls into one of the following categories:

1. **PBM embedded inside neural networks (NNs)**: Examples of these are embedding a differentiable convex optimization solver [3] or a rigid body simulator [4] in a neural network. A common challenge with these methods is that they are computationally expensive not only for training but also for inference.
2. **Reduced Order Model (ROM)**: The ROM approach [5] involves projecting complex partial differential equations (PDEs) onto a reduced dimensional space based on the singular value decomposition of the offline high fidelity simulation snapshots resulting in a set of ordinary differential equations which are fast to solve. Xiang et al. [6] applied ROM to model heat transfer in a battery pack of an electric vehicle, while Georgaka et al. [7] applied it to model turbulent heat transfer problems, and Li et al.

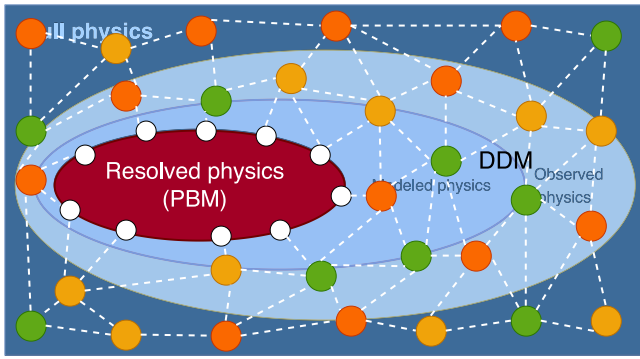


Fig. 3. Hybrid analysis and modeling: It maximizes the utilization of the well known PBM while correcting for the unknown using DDM. In the CoSTA, PBM is described by partial differential equations and DDM is a DNN.

[8] modeled steady-state and transient heat transfer in fractured geothermal reservoir. A recent review conducted by Ahmed et al. [9] gives a detailed overview of the various ROM approaches. Despite the huge potential ROM holds, their development requires knowledge of the equation governing the process to be modeled.

3. **Physics-informed neural networks (PINN):** The work of Raissi et al. [10] involves penalizing the cost function of the neural network with the residual of governing equations representing physical laws. Penwarden et al. [11] propose a particular multifidelity approach applied to PINNs that exploits low-rank structure. In the context of heat transfer, PINN has been used by He et al. [12] for solving both direct and inverse heat conduction problems. The penalization of the cost function can result in challenges during the optimization process due to the increased complexity of the cost function. Moreover, the exact form of the governing equations is a prerequisite for the method to work well.
4. **Data-driven equation discovery:** Sparse regression based on l_1 regularization [13,14] and symbolic regression based on gene expression programming have been shown to be very effective in complex equation discovery directly from data [15]. Xu et al. [16] demonstrated deep-learning based discovery of partial differential equations in integral form from sparse and noisy data. However, the limitations of this approach are that either large number of additional features are required to be handcrafted (in case of sparse regression) based on prior knowledge or the resulting models are unstable and prone to overfitting (in case of symbolic regression based on gene expression programming). In the case of using deep learning, interpretability remains elusive.
5. **Physics-guided neural network (PGNN):** This concept has been recently introduced to improve the training and predictions of deep neural networks (DNN). Partial knowledge, prior information, or results from highly simplified (and hence incomplete) models are injected into an intermediate hidden layer of the neural network [17]. The injection helps in improving accuracy, reducing model uncertainty and enabling more robust training. The approach has been used to combine information from simplified analytical models or low-fidelity models with noisy data obtained either from experiments or high-fidelity simulations through a neural network. It has been shown that this multi-fidelity information fusion framework produces physically consistent models that achieve better generalizability than purely DDM [18,19]. However, since PGNN is just a special kind of

$$\boxed{\text{PBM}} \quad \tilde{\mathcal{N}} \hat{u} = \tilde{f} + \boxed{\text{DDM}} \quad \hat{\sigma}_{\text{NN}} \\ \text{CoSTA}$$

Fig. 4. CoSTA combines PBM and DDM into a unified model by adding a DNN-generated corrective source term to the governing equation of the PBM.

NN, misbehavior of the neural network in unseen conditions and lack of interpretability may still be an issue in high-stakes application.

A more comprehensive list of hybrid modeling approaches and their applications can be found in [20–22]. From a careful evaluation of these modeling approaches it becomes apparent that most of the HAM approaches mentioned above have some shortcomings. Some of the approaches, like PBM embedded inside NNs, tend to be computationally expensive. Others, such as ROM and PINN, require the exact form of the equations governing the physics to be modeled. Data-driven equation discovery based on symbolic regression can be unstable and not fit for interpretation, while PGNN also offers limited interpretability and few opportunities for NN sanity checks. Another observation in our literature survey was that a disproportionately large percentage of work on hybrid modeling was found in more physics and engineering oriented journals and lesser in the data-driven machine learning journals. These observations have motivated our recent work [23] on a different HAM approach – the **Corrective Source Term Approach (CoSTA)** – where a **PBM is augmented with a data-driven component**. More specifically, a DNN-generated corrective source term is added to the (discretized) governing equation(s) of the PBM such as to correct any errors present in the original PBM, as illustrated in Fig. 4. These errors may e.g. stem from partial knowledge, discretization, and/or inaccurate parameter estimation. An important difference between CoSTA and the other HAM approaches discussed above is that CoSTA utilizes PBM to the greatest extent possible.

In [23], CoSTA was demonstrated to work for simple, one-dimensional heat transfer problems. In the current work we extend the work to two dimensions and take a closer look at the interpretability of the DNN-generated source term. The main contribution of this work can be enumerated as follows:

- Provide a brief presentation of the approach's underlying mathematical foundation
- Apply the approach to model a wide variety of two-dimensional heat diffusion phenomena
- Demonstrate and discuss how the DNN-generated correction term can be interpreted in a physics context, thereby increasing the explainability and reliability of the approach.

In Section 2, we present a profound mathematical foundation of the approach. We then continue with a discussion on heat diffusion modeling and its importance in Section 3. Physics-based and data-driven heat diffusion models are presented in Sections 3.2 and 3.3, respectively. In Section 3.4, we explain in detail how to combine these models using the proposed hybrid approach, as illustrated in Fig. 4.

Section 4 is devoted to explaining the setup of our numerical experiments – including the manufactured solutions considered, our DNN architecture and hyperparameter choices, and our data generation, training and testing procedures. Our experimental results are presented and discussed in Section 5 before the article is concluded in Section 6 with a brief summary and an outlook on future work.

2. Theory

This section briefly presents the mathematical justification of the hybrid approach, henceforth called the Corrective Source Term Approach (CoSTA), originally introduced in [23,24] for modeling systems governed by linear partial differential equations (PDEs). The presentation largely follows Blakseth [24].

First, suppose we want to solve the following general problem, defined on a domain Ω with boundary $\partial\Omega$:

$$\mathcal{N}_\Omega u = f \quad \text{in } \Omega, \quad (1)$$

$$\mathcal{N}_{\partial\Omega} u = g \quad \text{on } \partial\Omega. \quad (2)$$

Here, u is the unknown of the problem, \mathcal{N}_Ω and $\mathcal{N}_{\partial\Omega}$ are linear operators² acting on u , and f and g are some functions. With this formulation, we also capture scenarios where there are multiple governing equations. In such scenarios, u is a vector, and f and g are vector-valued functions.

Assume now that we have a PBM designed to predict u , and let \tilde{u} denote the PBM's prediction of the true solution u .

If $\tilde{u} \neq u$, there is some error in the PBM, and this error must stem from one or more of the following sources:

1. The true function f in Eq. (1) is unknown, so it is approximated by \tilde{f} .
2. The true operator \mathcal{N}_Ω in Eq. (1) is unknown, so it is approximated by $\tilde{\mathcal{N}}_\Omega$.
3. The true function g in Eq. (2) is unknown, so it is approximated by \tilde{g} .
4. The true operator $\mathcal{N}_{\partial\Omega}$ in Eq. (2) is unknown, so it is approximated by $\tilde{\mathcal{N}}_{\partial\Omega}$.
5. A combination of the above.
6. The true governing equation (1) and the true boundary conditions (2) are known, but cannot be solved analytically. To obtain a prediction \tilde{u} , we must therefore solve some approximation of the true system constituted by Eqs. (1) and (2), which effectively puts us in one of the other cases. For example, one could approximate the true operator \mathcal{N}_Ω with some numerical operator \mathcal{N}_{num} e.g. based on finite-difference approximations, which is equivalent to Case 2.

We observe that Cases 3 and 4 are analogous to Cases 1 and 2 because $\mathcal{N}_{\partial\Omega}$ and g play exactly the same roles in Eq. (2) as \mathcal{N}_Ω and f do in Eq. (1).

Since Case 6 is also mathematically equivalent to one of the other cases, it suffices to consider Cases 1 and 2, and combinations thereof.

Suppose now that the PBM-predicted solution \tilde{u} is given as the solution of the following system:

$$\tilde{\mathcal{N}}_\Omega \tilde{u} = \tilde{f} \quad \text{in } \Omega, \quad (3)$$

$$\mathcal{N}_{\partial\Omega} \tilde{u} = g \quad \text{on } \partial\Omega. \quad (4)$$

This formulation encompasses both Case 1 (for $\tilde{\mathcal{N}}_\Omega = \mathcal{N}_\Omega$ and $\tilde{f} \neq f$), Case 2 (for $\tilde{\mathcal{N}}_\Omega \neq \mathcal{N}_\Omega$ and $\tilde{f} = f$), and combinations thereof (for $\tilde{\mathcal{N}}_\Omega \neq \mathcal{N}_\Omega$ and $\tilde{f} \neq f$). Furthermore, suppose we modify the system above by adding a source term $\hat{\sigma}$ to Eq. (3), and let the solution of the modified system be denoted \hat{u} . Then, the modified system reads

$$\tilde{\mathcal{N}}_\Omega \hat{u} = \tilde{f} + \hat{\sigma} \quad \text{in } \Omega, \quad (5)$$

$$\mathcal{N}_{\partial\Omega} \hat{u} = g \quad \text{on } \partial\Omega. \quad (6)$$

and the following theorem holds.

² For u to be uniquely defined, $\mathcal{N}_{\partial\Omega}$ must be the unity mapping along a portion of $\partial\Omega$ of length greater than zero.

Theorem. Let \hat{u} be a solution of Eqs. (5) and (6), and let u be a solution of Eqs. (1) and (2). Then, for all operators $\tilde{\mathcal{N}}_\Omega$, \mathcal{N}_Ω , $\tilde{\mathcal{N}}_{\partial\Omega}$ and $\mathcal{N}_{\partial\Omega}$ and all functions f , \tilde{f} , g and \tilde{g} such that \hat{u} and u are uniquely defined, there exists a function $\hat{\sigma}$ such that $\hat{u} = u$.

Proof. Define the residual r of the PBM's governing equation (3) as³

$$r = \tilde{\mathcal{N}}_\Omega u - \tilde{f}. \quad (7)$$

If we set $\hat{\sigma} = r$ in Eq. (5), we then obtain

$$\begin{aligned} \tilde{\mathcal{N}}_\Omega \hat{u} &= \tilde{f} + \hat{\sigma} \\ &= \tilde{f} + \tilde{\mathcal{N}}_\Omega u - \tilde{f} \\ &= \tilde{\mathcal{N}}_\Omega u \\ \implies \hat{u} &= u \quad \blacksquare \end{aligned}$$

The theorem above proves that, for any error in the PBM's governing equation (3), there always exists a corrective source term $\hat{\sigma}$ which we can add to that equation such that the solution \hat{u} of the modified governing equation (5) is equal to the true solution u . Furthermore, any error in the PBMs boundary conditions (Eq. (4)) can be corrected analogously, since Eqs. (5) and (4) have the same functional form. Thus, the true solution u of the true governing equations can always be retained by modifying an erroneous PBM with a corrective source term. This observation is the principal theoretical justification of CoSTA. It is worth pointing out that, so far, we have not made any assumption regarding the operators \mathcal{N}_Ω and $\mathcal{N}_{\partial\Omega}$ except that they are linear. Hence, the approach should be applicable across a wide array of physical problems that can be cast in the form of the above equations.

The broad *applicability* of the approach is not to be confused with its *generalizability*, which is its ability to provide accurate predictions for previously unseen states of some particular system. The case study presented in the following is aimed at demonstrating the approach's generalizability.

3. Heat diffusion modeling

To demonstrate the potential of CoSTA, we choose to study two dimensional heat diffusion problems. The main motivation for choosing such problems is two-fold. Firstly, temperature can give insight into a wide variety of physical phenomena.⁴ Secondly, cost-effective and non-intrusive measurement techniques (e.g. based on thermal cameras) exist to make high-resolution spatio-temporal temperature measurements. In real-world applications, such techniques can be used to obtain the reference data needed for training the DNN used by CoSTA.

In this following sections, we describe the PBM, DDM and CoSTA models used in our numerical experiments on 2D heat diffusion. These models are presented in Sections 3.2, 3.3 and 3.4, respectively. But first, we shall briefly describe the heat equation in Section 3.1.

³ Note that our definition is in some sense opposite of common practice; we have defined the residual by inserting the true solution into the approximate equation rather than inserting the approximate solution into the true equation. The latter is the conventional approach, and is used e.g. in truncation error analysis [25, chapter 8]. The reason for our choice is two-fold: (1) It yields the simplest proof of the theorem. (2) When observing a real-world system, it is often easier to measure its state than to find the exact governing equation describing said state.

⁴ Trivial examples include using temperature to evaluate the power output of a heater, or to indicate an impending malfunction due to overheating.

3.1. The heat equation

The heat equation, which describes heat conduction through solid materials, can be written as

$$\int_V \rho c_V \frac{\partial T}{\partial t} dV = \int_{\partial V} (k \nabla T) \cdot \hat{\mathbf{n}} dA + \int_V P dV \quad (8)$$

for a stationary system with volume V , surface ∂V , surface unit normal $\hat{\mathbf{n}}$, density ρ , specific heat capacity at constant volume c_V , conductivity k , internal heat generation rate P and temperature T . We take Eq. (8) to be the true governing equation for all systems considered in the present work. Comparing with the general formulation used in Section 2, Eq. (8) corresponds to Eq. (1) with

$$u = T, \quad \mathcal{N}_\Omega u = \int_V \rho c_V \frac{\partial T}{\partial t} dV - \int_{\partial V} (k \nabla T) \cdot \hat{\mathbf{n}} dA \quad (9)$$

and

$$f = \int_V P dV. \quad (10)$$

To have a complete formulation of the system at hand, we also need to formulate the boundary conditions (BCs) of the system. In this work, we only consider Dirichlet BCs, which means that the temperature at the domain boundary is specified by some function(s). For the 2D systems considered in this work, the Dirichlet BCs can be formulated as

$$\begin{aligned} T(x_e, y, t) &= T_e(y, t), & T(x_w, y, t) &= T_w(y, t), \\ T(x, y_n, t) &= T_n(x, t), & T(x, y_s, t) &= T_s(x, t), \end{aligned} \quad (11)$$

where the subscripts e, w, n and s denote quantities evaluated at, respectively, the eastern (right), western (left), northern (upper) and southern (bottom) domain boundaries, and T_e, T_w, T_n and T_s are the functions specifying the boundary temperature. Again comparing with Section 2, we see that Eq. (11) is equivalent to Eq. (2) with $u = T$, $\mathcal{N}_{\partial\Omega}$ as the unity operator, and g being equal to T_e, T_w, T_n or T_s depending on whether we are on the eastern, western, northern or southern part of $\partial\Omega$.

3.2. Physics-based modeling

We now want to obtain a PBM for Eqs. (8) and (11). By limiting ourselves to 2D systems and assuming k, ρ and c_V to be constant,⁵ we are able to rewrite Eq. (8) as

$$\begin{aligned} \int_{y_s}^{y_n} \int_{x_w}^{x_e} \frac{\partial T}{\partial t} dx dy &= \kappa \left(\left(\frac{\partial T}{\partial x} \right)_e - \left(\frac{\partial T}{\partial x} \right)_w \right. \\ &\quad \left. + \left(\frac{\partial T}{\partial y} \right)_n - \left(\frac{\partial T}{\partial y} \right)_s \right) \\ &\quad + \int_{y_s}^{y_n} \int_{x_w}^{x_e} \sigma dx dy, \end{aligned} \quad (12)$$

where $\kappa = k/(\rho c_V)$ and $\sigma = P/(\rho c_V)$. Eq. (12) can be solved numerically using the Implicit Euler FVM, which can be expressed on the following matrix form for two successive time levels n and $n+1$:

$$\mathbb{A} \mathbf{T}_p^{n+1} = \mathbf{b}(\mathbf{T}_p^n). \quad (13)$$

For a domain that is discretized with N_j grid cells in the x -direction and N_i grid cells in the y -direction, \mathbb{A} is a banded $(N_j N_i \times N_j N_i)$ -matrix with five non-zero diagonals, while \mathbf{T}_p and \mathbf{b} are $N_j N_i$ -dimensional vectors. The components of \mathbf{T}_p describe

the temperature at the grid cell centers, as predicted by the PBM. The components are ordered such that the first N_j components describe the temperature at bottom-most row of cell centers (from left to right), the subsequent N_j components correspond to the second row from the bottom (still from left to right), and so on. Precise definitions of \mathbb{A} and \mathbf{b} can be found in [24]. Here, we highlight that \mathbb{A} depends on the conductivity k , while \mathbf{b} depends on the heat generation rate P and the system's boundary conditions (cf. Eq. (11)) in addition to the predicted temperature distribution at the old time level n , \mathbf{T}_p^n . Comparing with Section 2, we see that Eq. (13) is equivalent to Eq. (3) with

$$\tilde{u} \leftrightarrow \mathbf{T}_p^{n+1}, \quad \tilde{\mathcal{N}}_\Omega \tilde{u} \leftrightarrow \mathbb{A} \mathbf{T}_p^{n+1}, \quad \text{and} \quad \tilde{f} \leftrightarrow \mathbf{b}. \quad (14)$$

In our numerical experiments, we use the LAPACK routine “?gesv” (accessed through the SciPy library) to solve the system (13). However, using a specialized solver for sparse, banded systems is advised for problems that are more computationally demanding than those considered herein.

3.3. Data-driven modeling

The crux of DDM is to learn physics directly from observational data. For transient systems, this can be done by training a DNN to learn a mapping between two subsequent observations of the system state. For the heat diffusion problems considered herein, we take an observed state to be a vector $\mathbf{T}_{\text{ref}}^n$ describing the true temperature at the center of the grid cells used to define the PBM, as described in Section 3.2. The mapping we want the DNN to learn is then given by

$$\begin{aligned} \text{DNN}_T : \mathbb{R}^{(N_j+2)(N_i+2)} &\rightarrow \mathbb{R}^{N_j N_i} \quad \text{such that} \quad \mathbf{T}_d^{n+1} = \mathbf{T}_{\text{ref}}^{n+1}, \\ \mathbf{T}_d^n &\mapsto \mathbf{T}_d^{n+1} \end{aligned} \quad (15)$$

where \mathbf{T}_d^{n+1} refers to the temperature profile predicted by the DDM at time level $n+1$. The dimensionality of the DNN output is lower than that of the DNN input because the input vector contains boundary temperatures while the output vector does not. Since we consider Dirichlet BCs where the boundary temperatures are known, we need not have the DNN predict the boundary temperatures. However, it is still potentially useful to include them as DNN input with the aim of making the DNN's learning task easier. In an effort to reduce notational complexity, we use the same notation to denote both vectors with and without boundary information. Furthermore, we will use the notation DNN_T to refer to both the mapping defined by Eq. (15) and any DNN trained to approximate that mapping.

Our reason for choosing DNN-based DDM over other applicable DDMs is that DNNs have the ability to approximate any non-linear mapping, as guaranteed by the universal approximation theorem. Notice also that if we know the true initial condition of a system, i.e., if we can set $\mathbf{T}_d^0 = \mathbf{T}_{\text{ref}}^0$, then we will have $\mathbf{T}_d^n = \mathbf{T}_{\text{ref}}^n \forall n \geq 0$ if the mapping (15) is learnt perfectly.⁶

To summarize, the DDM used in this work is a DNN denoted DNN_T which is trained to predict \mathbf{T}_d^{n+1} given $\mathbf{T}_{\text{ref}}^n$ for any time level n . During testing, the output of DNN_T at time level n is used as its input at time level $n+1$, i.e. $\mathbf{T}_d^{n+1} = \text{DNN}_T(\mathbf{T}_d^n)$. Since we use $\mathbf{T}_d^0 = \mathbf{T}_{\text{ref}}^0$, we will have $\mathbf{T}_d^n = \mathbf{T}_{\text{ref}}^n \forall n \geq 0$ if DNN_T is perfectly trained.

⁵ In our numerical experiments, we consider scenarios where the assumption of constant k does not hold. Including constant k thereby allows us to synthesize modeling error in the PBM.

⁶ It should be noted that, depending on the discretization used, Eq. (15) may not constitute a well-defined mapping. Particular care should be taken for systems governed by so-called hyperbolic PDEs which permit discontinuous solutions. However, Eq. (15) is a well-defined mapping for the cases considered in our numerical experiments.

3.4. Hybrid analysis and modeling with CoSTA

In this section, we will briefly explain how we use CoSTA to model 2D heat diffusion. The PBM on which we base our CoSTA model is the Implicit Euler FVM described in Section 3.2. In Section 2, we established that the ideal corrective source term is given generally as the residual defined in Eq. (7). For the Implicit Euler FVM, we recall that $\tilde{N}_\Omega \leftrightarrow \mathbb{A}$ and $f \leftrightarrow \mathbf{b}$. Moreover, we have $u \leftrightarrow \mathbf{T}_{\text{ref}}^{n+1}$, such that the ideal corrective source term for the Implicit Euler FVM reads

$$\hat{\sigma}^{n+1} = \mathbb{A} \mathbf{T}_{\text{ref}}^{n+1} - \mathbf{b}(\mathbf{T}_{\text{ref}}^n). \quad (16)$$

As in Section 2, we use the corrective source term to define a modified governing equation whose solution is exactly equal to the reference solution at all grid nodes and at all time levels. We use a subscript _h (for “hybrid analysis and modeling”) to denote the solution of the modified system, and we write the modified system as

$$\mathbb{A} \mathbf{T}_h^{n+1} = \mathbf{b}(\mathbf{T}_h^n) + \hat{\sigma}^{n+1}. \quad (17)$$

For a posteriori analyses, these equations can be used directly. However, for a priori predictions, $\mathbf{T}_{\text{ref}}^{n+1}$ is unknown. We therefore use a DNN-generated corrective source term $\hat{\sigma}_{\text{NN}}^{n+1}$ to approximate the true corrective source term $\hat{\sigma}^{n+1}$. As input to the DNN generating $\hat{\sigma}_{\text{NN}}^{n+1}$, we use a predictor $\tilde{\mathbf{T}}_h^{n+1}$ defined by

$$\mathbb{A} \tilde{\mathbf{T}}_h^{n+1} = \mathbf{b}(\mathbf{T}_h^n), \quad (18)$$

with \mathbb{A} and \mathbf{b} defined as in Eq. (12). Our choice of DNN input was inspired by predictor–corrector schemes used for numerical integration. We make no claim that this choice is optimal, but observe that it has worked well in our numerical experiments. With this choice of input, we want to train the DNN of the CoSTA model to approximate the following mapping:

$$\text{DNN}_\sigma : \mathbb{R}^{(N_j+2)(N_i+2)} \rightarrow \mathbb{R}^{N_j \cdot N_i} \quad \text{such that} \quad \hat{\sigma}_{\text{NN}}^{n+1} = \hat{\sigma}^{n+1}. \quad (19)$$

$$\tilde{\mathbf{T}}_h^{n+1} \mapsto \hat{\sigma}_{\text{NN}}^{n+1}$$

As for the DDM mapping (15), the dimensionality reduction originates from the use of Dirichlet BCs. Furthermore, we use the notation DNN_σ to refer to both the mapping (19) and any DNN trained to approximate that mapping.

4. Experimental setup and procedures

4.1. Data generation

We use the method of manufactured solutions (MMS) to generate data for our numerical experiments. Our motivation for using synthetic data generated using MMS instead of real data, is to make the analysis of CoSTA’s accuracy and interpretability as rigorous as possible. Real data inevitably contains some noise, which would make it difficult to differentiate the models’ accuracy in scenarios where several models perform well. Moreover, when the reference data is noisy, the true corrective source term $\hat{\sigma}$ (as defined by Eq. (5)) is not known precisely. This would make it difficult to assess our hypothesis that the DNN-generated corrective source term $\hat{\sigma}_{\text{NN}}$ can be interpreted to obtain useful information. In the present work, we feel it was important to keep the number of error sources to a minimum, such as to keep the analysis as straight-forward as possible.

The main concept of MMS is to prescribe some convenient but otherwise arbitrary function as the solution of the governing equation (the heat Eq. (8) in our case). All of the parameters of the equation except for one (typically the heat generation rate P for the heat equation) are also prescribed. The final parameter is then calculated by inserting the prescribed solution

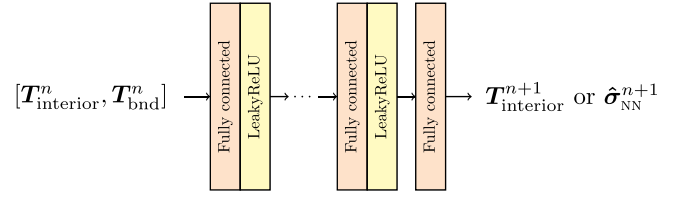


Fig. 5. The fully connected DNN architecture used in the numerical experiments of the present work. Temperatures at both the domain interior and the boundary are given as model input. The data is processed by a number of fully connected layers with LeakyReLU activation functions. A fully connected layer without activation functions is used to generate the final output. For DDM, the output is the temperature in the domain interior⁷ at the subsequent time level. For CoSTA, the output is the corrective source term $\hat{\sigma}_{\text{NN}}^{n+1}$. Source: Adapted from [23].

and parameters into the governing equation. This way, it is easy to obtain analytical solutions for any governing equation. Thus, MMS is a powerful tool for generating synthetic data for numerical experiments without the use of any expensive high-fidelity solvers.

Due to the integrals, Eq. (8) is not convenient for use with MMS. Therefore we use instead the so-called differential form of the heat equation when generating data for our numerical experiments. For smooth temperature profiles, the differential form is equivalent to Eq. (8), and it reads

$$\rho c_v \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k \frac{\partial T}{\partial y} \right) + P. \quad (20)$$

It is clear from the equation that given any P independent of the third dimension (which is the case for all the cases attempted), the heat transfer phenomena will be completely two-dimensional. Therefore, from here onwards, we will assume that the third dimension is of unit thickness. Our data generation procedure is then to prescribe T , k , ρ and c_v in the equation above and calculate the P required for the equation to be satisfied. For simplicity we always prescribe ρ and c_v to unity in this work.

We consider a total of four different manufactured solutions T_{ref} as listed in Table 1. The corresponding chosen k and calculated P are also included in the table. All the manufactured solutions are parametrized by a parameter α which allows us to generate several time series using the same manufactured solutions. We consider a total of 22 different α -values, such that we get 22 unique time series for each manufactured solutions. We emphasize that this way of choosing uniformly spaced values of α is not optimal. If we were to use resource-intensive experimentation for data generation then a better approach would have been to use concepts from Design of Experiments to minimize the number of experiments while still generating informative data.

As shown in Table 2, 16 of these were used for DNN training, 2 were used for DNN validation, and 4 were used for model testing. Each time series was discretized using $N_t = 5001$ time levels on the temporal domain $[0 \text{ s}, 5 \text{ s}]$ and 20×20 grid cells on the spatial domain $[0 \text{ m}, 1 \text{ m}] \times [0 \text{ m}, 1 \text{ m}]$. Furthermore, a unit thickness for the plate has been assumed. All models operate on flattened data, meaning that any discretized 2D temperature field is represented by a 1D vector in the models.

4.2. DNN setup and training routines

The fully-connected DNN architecture we use is illustrated in Fig. 5, and our hyperparameter choices are listed in Table 3. To be

⁷ We assume the BCs are known, so there is no need for the DNN to predict boundary temperatures.

Table 1

Manufactured solutions T_{ref} used for our experiments. Each solution is taken to be defined on the spatial domain $[0 \text{ m}, 1 \text{ m}] \times [0 \text{ m}, 1 \text{ m}]$ and the temporal domain $[0 \text{ s}, 5 \text{ s}]$. P and k are given in their respective SI units, while T is given in degrees Celsius.

Label	$T_{\text{ref}}(x, y, t; \alpha)$	$P(x, y, t; \alpha)$	$k(x, y, t; \alpha)$
2P1	$t + 0.5\alpha(x^2 + y^2) + x$	$(1 - 2\alpha)$	1
2P2	$1 + \sin(2\pi t + \alpha) \cos(2\pi x) \cos(2\pi y)$	$2\pi \cos(2\pi x) \cos(2\pi y) (\cos(2\pi t + \alpha) + 4\pi \sin(2\pi t + \alpha))$	1
2k1	$t + \alpha x + y^2$	$-(1 + \alpha + 2x + 4y)$	$1 + x + y$
2k2	$\alpha + (t + 1) \cos(2\pi x) \cos(4\pi y)$	$\cos(2\pi x) \cos(4\pi y) (1 + 40\pi^2(t + 1) (1 + \sin(1\pi x) \sin(4\pi y)))$	$2 + \sin(2\pi x) \sin(4\pi y)$

Table 2

Parametrization: Selection of α -values corresponding to the training, validation and testing time series used in our experiments. Note that in the test set $\mathcal{A}_{\text{test}}$, two values of $\alpha = -0.5$ and 2.5 correspond to extrapolation scenarios while $\alpha = 0.7$ and 1.5 correspond to interpolation scenarios.

Purpose	Set of α -values	Symbol
Training	$\{0.1, 0.2, \dots, 2.0\} \setminus \{0.7, 0.8, 1.1, 1.5\}$	$\mathcal{A}_{\text{train}}$
Validation	$\{0.8, 1.1\}$	\mathcal{A}_{val}
Testing	$\{-0.5, 0.7, 1.5, 2.5\}$	$\mathcal{A}_{\text{test}}$

Table 3

The DNN hyperparameters used in our experiments.

Parameter	Value
Loss function	MSE
Learning rate	$1e-5$
Optimizer	Adam
Batch size	32
# hidden FC layers	4
Hidden FC layer width	80
LeakyReLU slope	0.01
Validation period	$1e2$
Overfit limit	20

compatible with our chosen spatial discretization, the DNN input and output layers must consist of 484 and 400 nodes respectively.⁸ The training procedures for the DNNs of DDM and CoSTA are illustrated in Fig. 6(a) for a single data example ($\mathbf{T}_{\text{ref}}^n, \mathbf{T}_{\text{ref}}^{n+1}$).

4.3. Testing

In each of our numerical experiments, we consider one of the manufactured solutions listed in Table 1, and attempt to replicate the four time series corresponding to $\alpha \in \mathcal{A}_{\text{test}}$. For each time series, we inform the models of the true initial condition and the true boundary conditions.

To synthesize modeling error in the PBM, we set $P = 0$ in the PBM when modeling Systems 2P1 and 2P2. However, for Systems 2k1 and 2k2, we inform the PBM of the true P . In these cases, modeling error is instead synthesized by the assumption of constant k . More specifically, we set $k = 1$ in PBM for all the numerical experiments considered herein. Our hypothesis is that the corrective source term in the CoSTA model will correct for the modeling error synthesized in the PBM, irrespective of the whether the error stems from an incorrect P or an incorrect k . We highlight that no modeling error is synthesized in the DDM model.

For assessing the quality of the three models' predictions, we use the relative ℓ_2 -norms

$$E_p = \frac{\|\mathbf{T}_p^n - \mathbf{T}_{\text{ref}}^n\|_2}{\|\mathbf{T}_{\text{ref}}^n\|_2}, \quad E_d = \frac{\|\mathbf{T}_d^n - \mathbf{T}_{\text{ref}}^n\|_2}{\|\mathbf{T}_{\text{ref}}^n\|_2}, \quad (21)$$

$$E_h = \frac{\|\mathbf{T}_h^n - \mathbf{T}_{\text{ref}}^n\|_2}{\|\mathbf{T}_{\text{ref}}^n\|_2},$$

⁸ For our chosen discretization, the number of grid cells is 400, and the number of boundary nodes is 84. Since the boundary conditions are known in advance they are omitted from the output layers.

where

$$\|\mathbf{v}\|_2 = \left(\sum_{i=1}^D v_i^2 \right)^{1/2} \quad (22)$$

for any D -dimensional vector \mathbf{v} . The training and testing procedures are illustrated in Fig. 6. Algorithmic representations of the training and testing are given in Algorithms 1 and 2 of Blakseth et al. [23].⁹

5. Results and discussion

In this section, we present and discuss the results of our four numerical experiments. The experiments concerning solutions 2P1 and 2P2, where modeling error due to an unknown P is the dominant error source in the PBM, are considered in Section 5.1. The experiments concerning solutions 2k1 and 2k2, where an incorrectly modeled k is the primary PBM error source, are considered thereafter in Section 5.2. Finally, the section is concluded by a discussion on the interpretability of the corrective source term in Section 5.3.

In Sections 5.1 and 5.2, the results are grouped in interpolation scenarios $\alpha \in \{0.7, 1.5\}$ and extrapolation scenarios $\alpha \in \{-0.5, 2.5\}$. We make a distinction between the interpolation and extrapolation scenarios during testing because data-driven models tend to do relatively better in interpolation compared to the extrapolation scenarios. This is due to the fact that the test data corresponding to interpolations are better represented by the training data. Since by design, PBMs do not differentiate between the two scenarios, it is expected that a hybrid approach will inherit this strength of the PBM, and consequently perform better than pure DDM in the extrapolation scenarios.

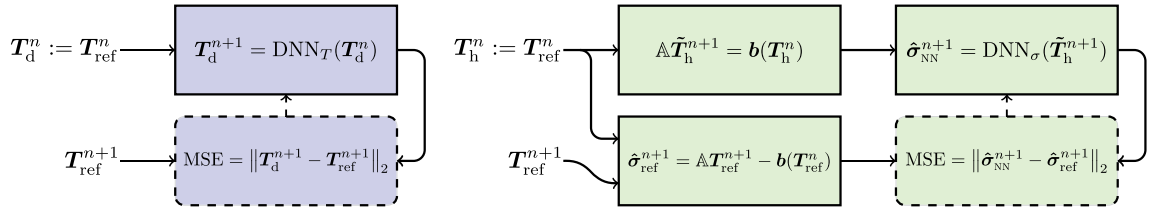
Both result sections begin with a discussion on the interpolation scenarios, while the extrapolation scenarios are considered thereafter. For each α -value and each manufactured solution, we display the temporal development of the ℓ_2 -errors E_p , E_d and E_h defined in Eq. (21) (cf. Figs. 7, 12, 17 and 22). Additionally, we also display the relative error fields $(\mathbf{T}_p^{N_t-1} - \mathbf{T}_{\text{ref}}^{N_t-1})/\mathbf{T}_{\text{ref}}^{N_t-1}$, $(\mathbf{T}_d^{N_t-1} - \mathbf{T}_{\text{ref}}^{N_t-1})/\mathbf{T}_{\text{ref}}^{N_t-1}$, and $(\mathbf{T}_h^{N_t-1} - \mathbf{T}_{\text{ref}}^{N_t-1})/\mathbf{T}_{\text{ref}}^{N_t-1}$, where all subtractions and divisions are applied component-wise.¹⁰

5.1. Experiments with unknown source term

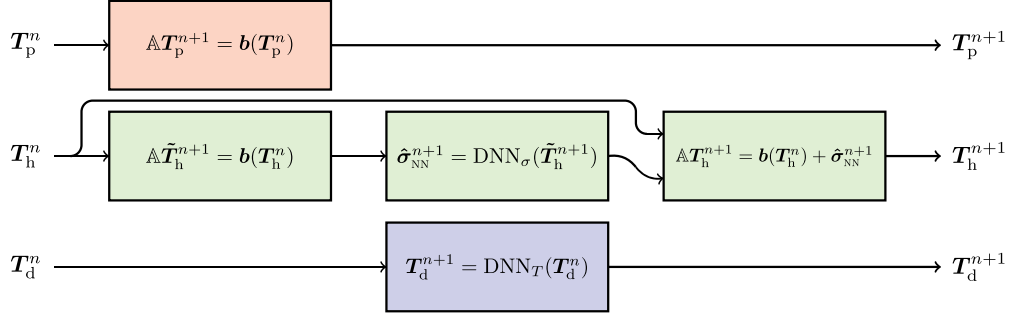
In this section, we consider two experiments where the source term P of the heat equation is assumed unknown. From a physical point of view, this can be interpreted as some unknown heating within the system. For example, P can correspond to an unknown power output of a heater in a room, to heat generated from electrical resistance in a system influenced by electrical currents, or to heat generated by friction inside a system with moving

⁹ Keep in mind that, while the procedures used here and in [23] are completely analogous, there are some technical difference. An obvious example is the definitions of the temperature vectors. Moreover, Blakseth et al. [23] use \hat{q} to refer to the heat generation rate P which we here denote P .

¹⁰ For these illustrations, we use the `imshow` function of Matplotlib, which interpolates the discrete differences to produce smooth error fields.



(a) Training procedures for the DNNs used in CoSTA/DDM (left) and HAM (right). Note that, for all n , we use the reference profile as input during training. As such, the models are only trained to make *local* (i.e. single-step) corrections. Moreover, they are not trained to recognize errors in their own predictions at previous time levels.



(b) Time stepping procedures for PBM (top), CoSTA/HAM (middle) and DDM (bottom). For all models, the output at one time level will be used as input at the subsequent time level.

Fig. 6. Training and time stepping procedures for the three modeling approaches PBM (red), DDM (blue) and HAM (green). Note that PBM is not included in (a) because it does not require any training.
Source: Figure adapted from [23].

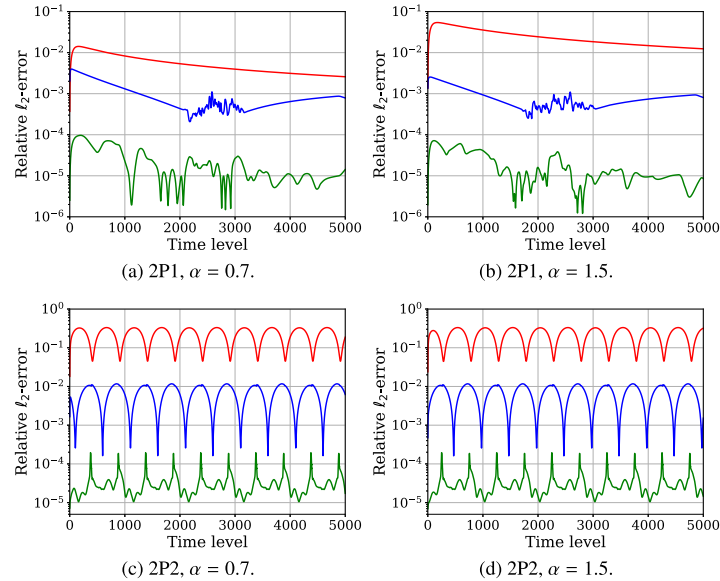


Fig. 7. Solutions 2P1 and 2P2, interpolation: Relative ℓ_2 -errors for $\alpha \in \{0.7, 1.5\}$ (— PBM, — DDM, — HAM).

components. The manufactured solutions studied in this section are Solutions 2P1 and 2P2 (cf. Table 1). We discuss the results for the interpolation scenarios ($\alpha \in \{0.7, 1.5\}$) first and the extrapolation scenarios ($\alpha \in \{-0.5, 2.5\}$) thereafter.

The results for Solutions 2P1 and 2P2 in the interpolation scenarios are shown in Figs. 7–11. From the temporal development of the models' relative ℓ_2 -errors (Fig. 7), we see that the models follow a clear hierarchy in terms of accuracy. In all four cases, the PBM is the least accurate model. The PBM's accuracy is especially poor for Solution 2P2, for which it produces relative ℓ_2 -errors of up to 30%. The DDM is the second most accurate model, producing relative ℓ_2 -errors which are, on average, roughly one order of magnitude smaller than those of the PBM. However, in

all cases, the CoSTA-based HAM model is by far the most accurate model. We observe that the addition of the DNN-generated corrective source term yields an increase in accuracy of roughly three orders of magnitude compared to the unmodified PBM. We also observe that the CoSTA-based HAM model generally outperforms the DDM by more than one order of magnitude. The CoSTA model and the DDM use the same DNN using the same hyperparameters and the same training regime, so this result must imply that using a PBM to account for some physics, as is done in the CoSTA model, is more efficient than using the DNN to account for all physics, as is done in the DDM model. We notice that this holds true even when the accuracy of the PBM itself is poor (cf. the results for Solution 2P2).

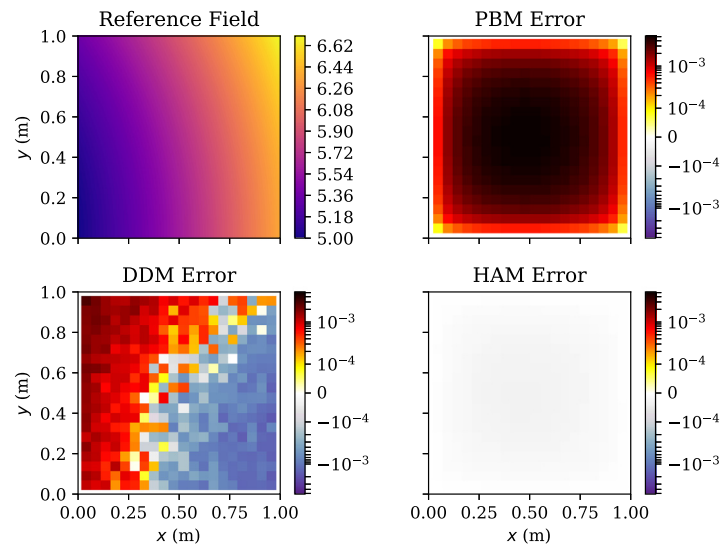


Fig. 8. Solution 2P1, $\alpha = 0.7$: Reference temperature field and relative ℓ_2 -errors of PBM, DDM and HAM at $t = 5$ s.

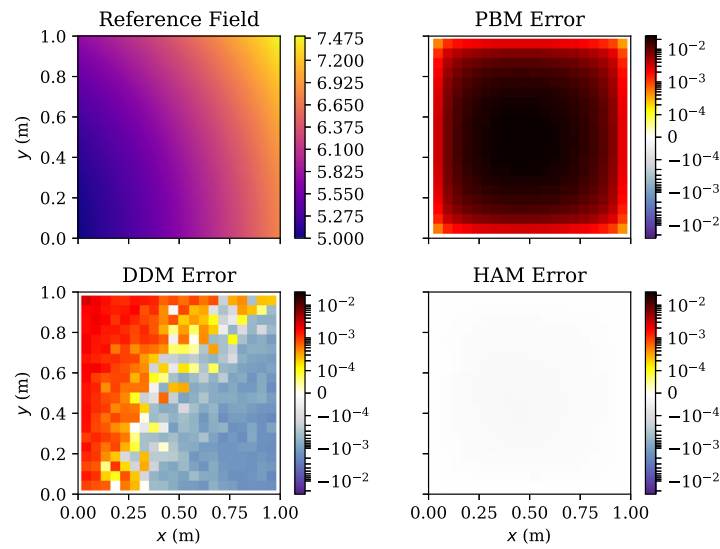


Fig. 9. Solution 2P1, $\alpha = 1.5$: Reference temperature field and relative ℓ_2 -errors of PBM, DDM and HAM at $t = 5$ s.

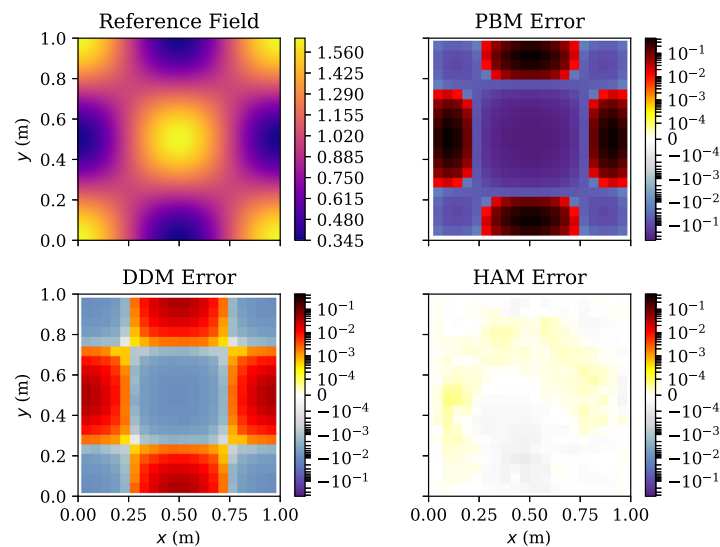


Fig. 10. Solution 2P2, $\alpha = 0.7$: Reference temperature field and relative ℓ_2 -errors of PBM, DDM and HAM at $t = 5$ s.

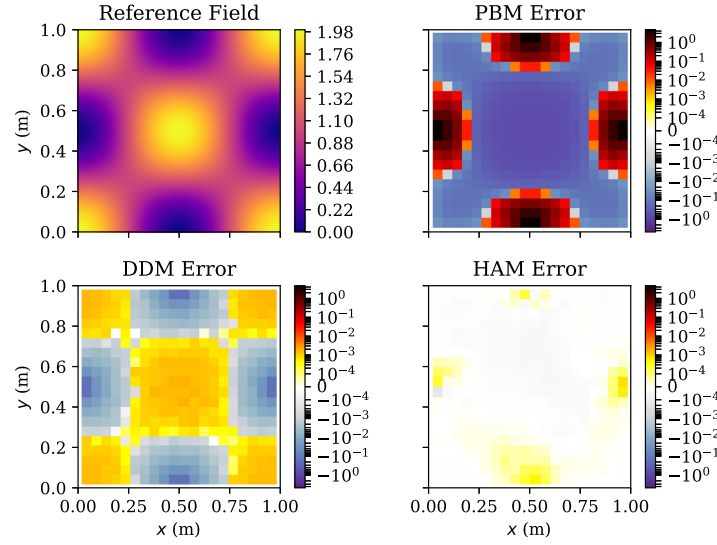


Fig. 11. Solution 2P2, $\alpha = 1.5$: Reference temperature field and relative ℓ_2 -errors of PBM, DDM and HAM at $t = 5$ s.

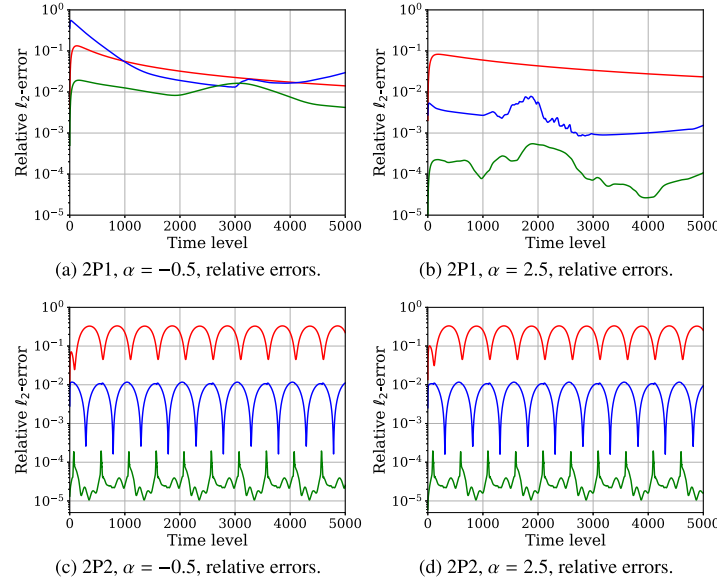


Fig. 12. Solutions 2P1 and 2P2, extrapolation: Relative ℓ_2 -errors for $\alpha \in \{0.7, 1.5\}$ (— PBM, — DDM, — HAM).

The extrapolation scenario results for Solutions 2P1 and 2P2 are shown in Figs. 12–16. By comparing the ℓ_2 -errors for the extrapolation scenarios (cf. Fig. 12) and the interpolation scenarios (cf. Fig. 7), we see that PBM is the most generalizable model, in the sense that its results for the interpolation and extrapolation scenarios are the most similar. In fact, there is no significant difference between the PBM's accuracy in the extrapolation scenarios and in the interpolation scenarios. However, it should be noted that the PBM is still the least accurate model overall. For Solution 2P2, DDM and HAM exhibit roughly the same level of accuracy in the interpolation scenarios, and this can be explained by the fact that Solution 2P2 is not qualitatively different in the extrapolation scenarios than in the interpolation scenarios. The qualitative difference between the scenarios is much greater for Solution 2P1, and this is clearly reflected in the DDM and HAM results. Both DDM and HAM suffer significant accuracy reduction going from interpolation to extrapolation for this solution. However, CoSTA-based HAM is still the most accurate model overall.

It is of particular interest to study the error fields corresponding to Solution 2P1, which are illustrated in Figs. 13 and 14. We observe that the PBM and HAM error fields are smooth and quite uniform throughout the spatial domain, while this is not the case for the DDM error fields. For $\alpha = -0.5$, the DDM prediction is significantly too hot in the top right corner. For $\alpha > 0$, that corner is the warmest corner, so this error illustrates a failure to generalize which is not observed for the PBM and CoSTA models. For $\alpha = 2.5$, we observe that the DDM error field is noisy. Since the DNN is trained using an ℓ_2 loss function which does not enforce smooth DNN output, this is not really surprising. However, it is worth noting that HAM is not affected by noise from the DNN to the same extent as the purely data-driven model.

Looking at the error curves for System 2P2 in Figs. 7 and 12, the curious reader may wonder why the error curves of the PBM are “out of phase” in comparison to the DDM and HAM curves. A careful examination reveals that the low-points of the PBM error curves correspond to temporal locations where $P \approx 0$. Since the

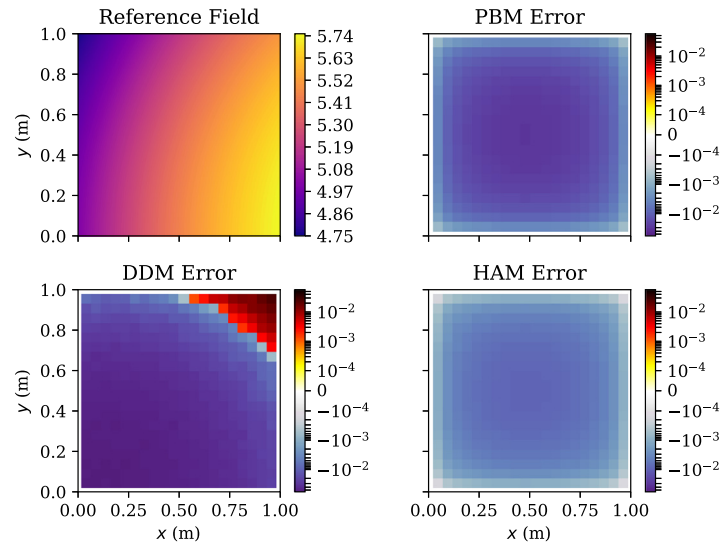


Fig. 13. Solution 2P1, $\alpha = -0.5$: Reference temperature field and relative ℓ_2 -errors of PBM, DDM and HAM at $t = 5$ s.

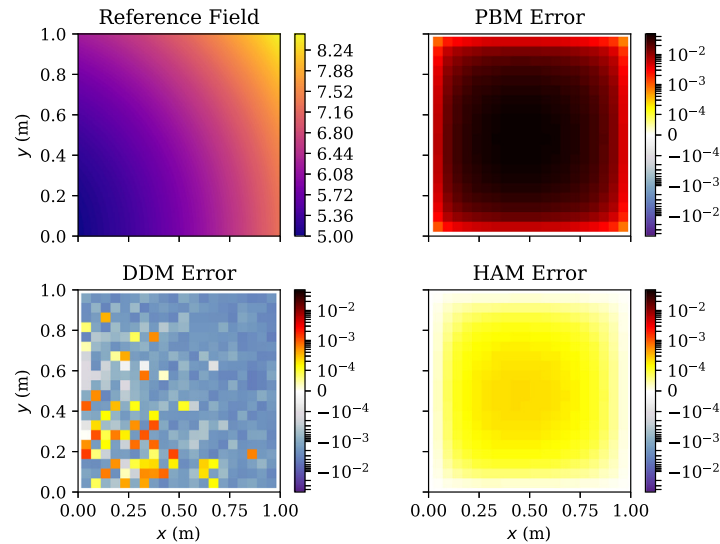


Fig. 14. Solution 2P1, $\alpha = 2.5$: Reference temperature field and relative ℓ_2 -errors of PBM, DDM and HAM at $t = 5$ s.

PBM assumes $P = 0$, this is sensible. Still, it may appear counter-intuitive that the DDM and HAM models perform worst for this simple situation. The reason is probably related to the training of the DNNs; most training examples seen by the DNNs correspond to curved temperature fields, so the DNNs are not well-trained for handling the special case $P \approx 0$ corresponding to virtually flat temperature fields. If the case $P \approx 0$ is of particular importance, data augmentation could be beneficial in obtaining more accurate predictions in this case.

5.2. Experiments with unknown conductivity

We now move on to our two experiments where the conductivity k is taken to be unknown. Such scenarios are found, for example, when studying composite systems, mixtures or other inhomogeneous systems. In such cases, accurately determining the thermal conductivity at all locations within the system can be forbiddingly challenging, or even unfeasible. In this section, we consider the two manufactured solutions 2k1 and 2k2 (cf. 1), whose corresponding conductivity profiles are linear and periodic in space, respectively. As in the previous section, we discuss

the interpolation scenarios first and the extrapolation scenarios thereafter.

The interpolation scenario results for Solutions 2k1 and 2k2 are shown in Figs. 17–21. As in the previous experiments, we see from Fig. 17 that the CoSTA-based HAM model is the most accurate model. On the whole, CoSTA is at least one order of magnitude more accurate than DDM, which is in turn at least one order of magnitude more accurate than PBM. The difference in accuracy is particularly striking for Solution 2k1, for which the DNN-generated source term of CoSTA yields an accuracy increase of roughly four orders of magnitude in comparison to the uncorrected PBM.

The results for Solutions 2k1 and 2k2 in the extrapolation scenarios $\alpha \in \{-0.5, 2.5\}$ are shown in Figs. 22–26. From the ℓ_2 -errors illustrated in Fig. 22, we see that CoSTA maintains its position as the most accurate model. We also observe that the DDM model apparently generalizes well to the scenario $\alpha = 2.5$ for Solution 2k1, almost matching the accuracy of CoSTA in that scenario. However, the DDM model is the least accurate model for the same solution with $\alpha = -0.5$, being more than one order of magnitude less accurate in the latter scenario than in

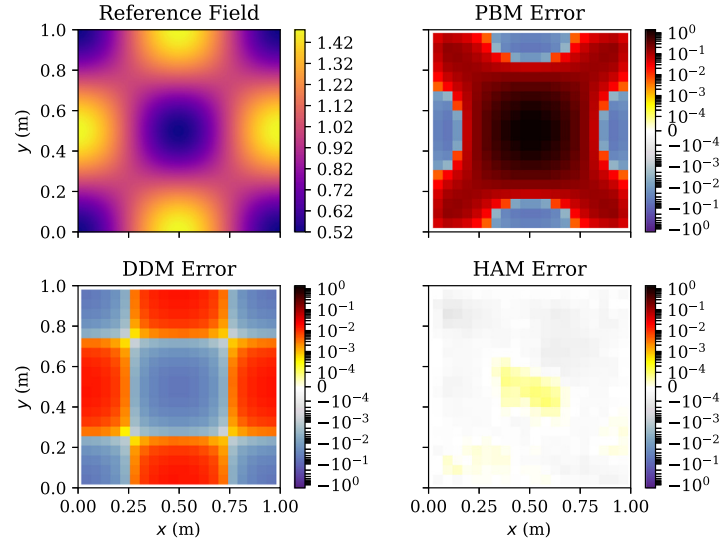


Fig. 15. Solution 2P2, $\alpha = -0.5$: Reference temperature field and relative ℓ_2 -errors of PBM, DDM and HAM at $t = 5$ s.

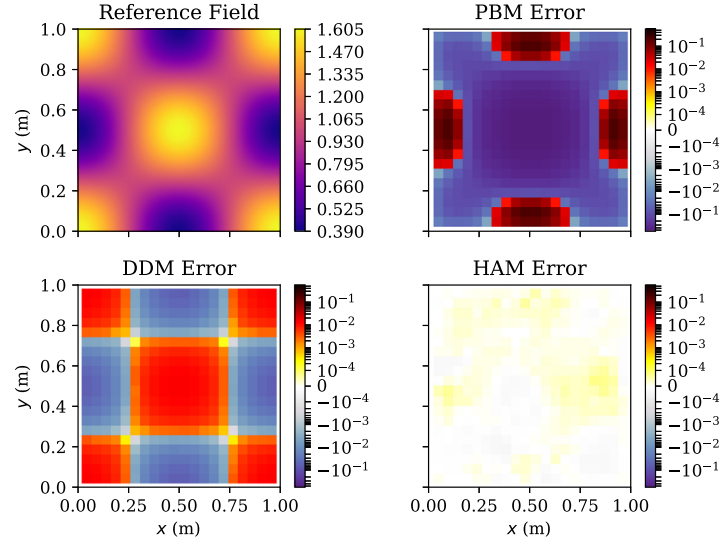


Fig. 16. Solution 2P2, $\alpha = 2.5$: Reference temperature field and relative ℓ_2 -errors of PBM, DDM and HAM at $t = 5$ s.

the former. On the other hand, CoSTA maintains the same level of accuracy in both scenarios, thereby exhibiting better generalization than DDM. This conclusion is further strengthened by the error fields shown in Fig. 23. From that figure, we see that the prediction of the CoSTA-based HAM model is qualitatively correct but somewhat too hot over the entire domain. However, the DDM prediction is decidedly too cold over most of the domain while being too warm for $x \gtrsim 0.75$ m. For the system at hand, there is nothing special occurring at this vertical line, so the error field indicates that the DDM predictions for this scenario are qualitatively incorrect and possibly even unphysical. The errors fields shown in Fig. 24 tell a similar story; the CoSTA prediction is too cold but otherwise qualitatively correct, while the DDM error field is noisy and has no clear connection to the reference temperature field.

The significant ℓ_2 -error drop observed for System 2k2 in Fig. 22 can possibly be explained by noting that, since the oscillation amplitude of T_{ref} increases with time, the importance of α (which defined the center of the oscillation) decreases with time. As such, the observed errors are consistent with the HAM

and DDM models being able to model the spatial oscillation well while having more difficulties modeling α accurately (with DDM struggling significantly more than HAM).

5.3. Interpretation of the corrective source term

Interpretability of the DNN-generated corrective source term has been highlighted as one of CoSTA's major strong-points. In this section, we aim to substantiate these claims regarding interpretability by demonstrating possible ways of interpreting the corrective source term for Solutions 2P2 and 2k1 with $\alpha = -0.5$ and $\alpha = 0.7$. We will also provide some general discussions related to interpretation of the corrective source term.

Let us first consider the case where we want to model a system with unknown heat generation rate P . Suppose we approximate the true P with $\tilde{P} = P - \epsilon_P$, where $\epsilon_P \neq 0$ is the error of the approximation. (For Solutions 2P1 and 2P2 considered in the numerical experiments above, we used $\tilde{P} = 0$, such that $\epsilon_P = P$.)

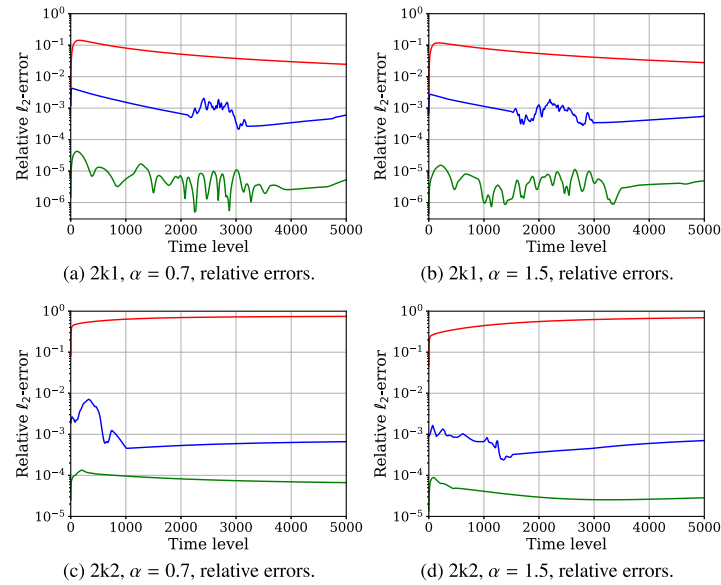


Fig. 17. Solutions 2k1 and 2k2, interpolation: Relative ℓ_2 -errors for $\alpha \in \{0.7, 1.5\}$ (— PBM, — DDM, — HAM).

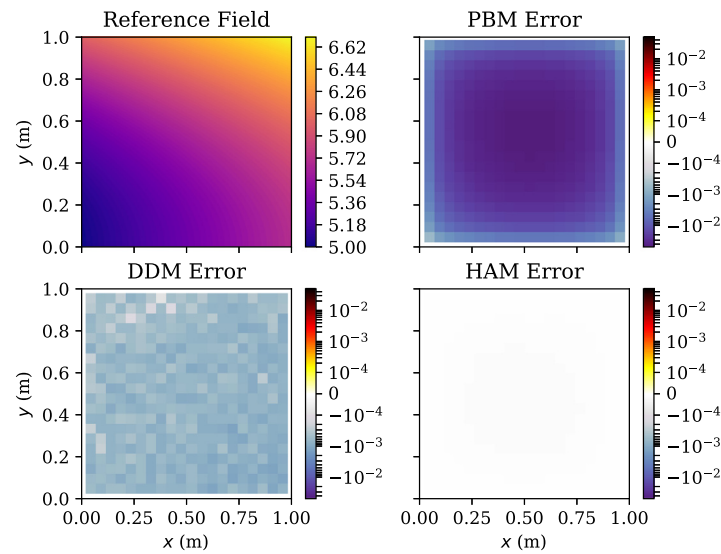


Fig. 18. Solution 2k1, $\alpha = 0.7$: Reference temperature field and relative ℓ_2 -errors of PBM, DDM and HAM at $t = 5$ s.

Inserting $\tilde{P} + \epsilon_p$ for P into Eq. (8) for 2D systems, we obtain

$$\begin{aligned}
 \int_V \rho c_V \frac{\partial T}{\partial t} dV &= \left(kA \frac{\partial T}{\partial x} \right)_e - \left(kA \frac{\partial T}{\partial x} \right)_w \\
 &+ \left(kA \frac{\partial T}{\partial y} \right)_n - \left(kA \frac{\partial T}{\partial y} \right)_s \\
 &+ \int_V P dV \\
 &= \left(kA \frac{\partial T}{\partial x} \right)_e - \left(kA \frac{\partial T}{\partial x} \right)_w \\
 &+ \left(kA \frac{\partial T}{\partial y} \right)_n - \left(kA \frac{\partial T}{\partial y} \right)_s \\
 &+ \int_V \tilde{P} dV + \int_V \epsilon_p dV.
 \end{aligned} \tag{23}$$

Following the discretization procedure used to derive the Implicit Euler FVM (13),¹¹ we can discretize the above equation as follows:

$$\mathbb{A} \mathbf{T}^{n+1} = \mathbf{b}(\mathbf{T}^n) + \Delta t \tilde{\sigma}_p. \tag{24}$$

Here, \mathbb{A} , \mathbf{T} and \mathbf{b} are defined as in Eq. (13), and $\tilde{\sigma}_p = \epsilon_p / (\rho c_V)$, where $\epsilon_p = [\epsilon_p(x_1, y_1), \dots, \epsilon_p(x_{N_j}, y_{N_j})]$ and $(x_1, y_1), \dots, (x_{N_j}, y_{N_j})$ are the grid nodes used to discretize the spatial domain. Comparing Eqs. (24) to (17), we can see that there is a clear connection between $\tilde{\sigma}_p$ and the corrective source term $\hat{\sigma}$ used in CoSTA. One might be tempted to simply write $\Delta t \tilde{\sigma}_p = \hat{\sigma}$, but this equality does not hold true in general. The reason for this is that $\hat{\sigma}$ accounts for *all* error in the PBM, while $\tilde{\sigma}_p$ accounts only for

¹¹ This discretization procedure is considered in detail in Section 2.2.3 of Blakseth [24].

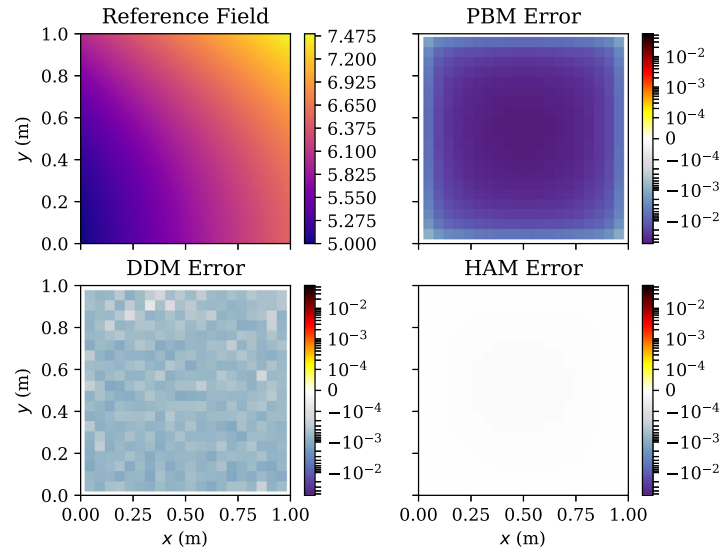


Fig. 19. Solution 2k1, $\alpha = 1.5$: Reference temperature field and relative ℓ_2 -errors of PBM, DDM and HAM at $t = 5$ s.

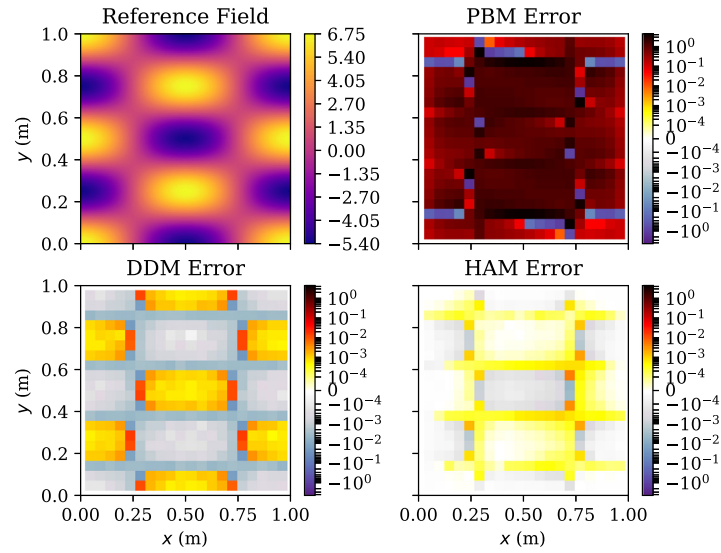


Fig. 20. Solution 2k2, $\alpha = 0.7$: Reference temperature field and relative ℓ_2 -errors of PBM, DDM and HAM at $t = 5$ s.

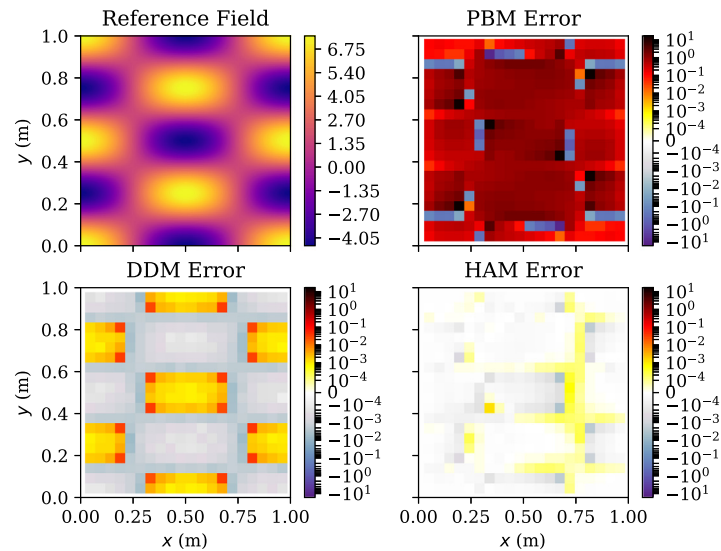


Fig. 21. Solution 2k2, $\alpha = 1.5$: Reference temperature field and relative ℓ_2 -errors of PBM, DDM and HAM at $t = 5$ s.

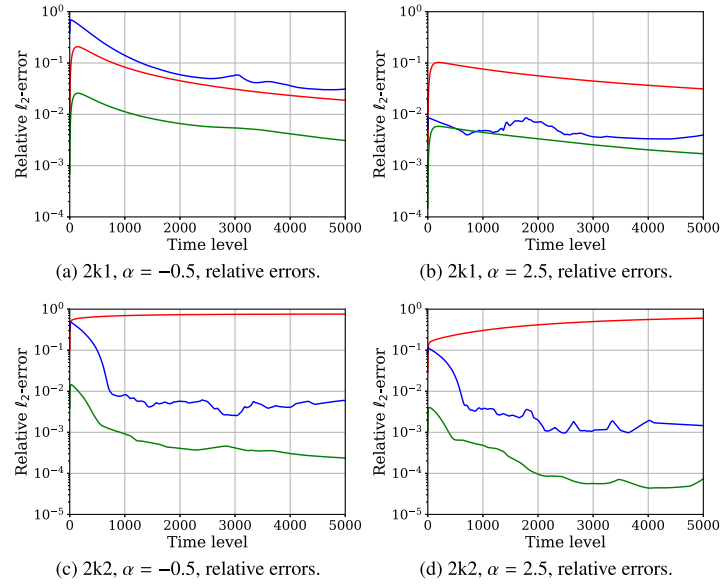


Fig. 22. Solutions 2k1 and 2k2, extrapolation: Relative ℓ_2 -errors for $\alpha \in \{0.7, 1.5\}$ (— PBM, — DDM, — HAM).

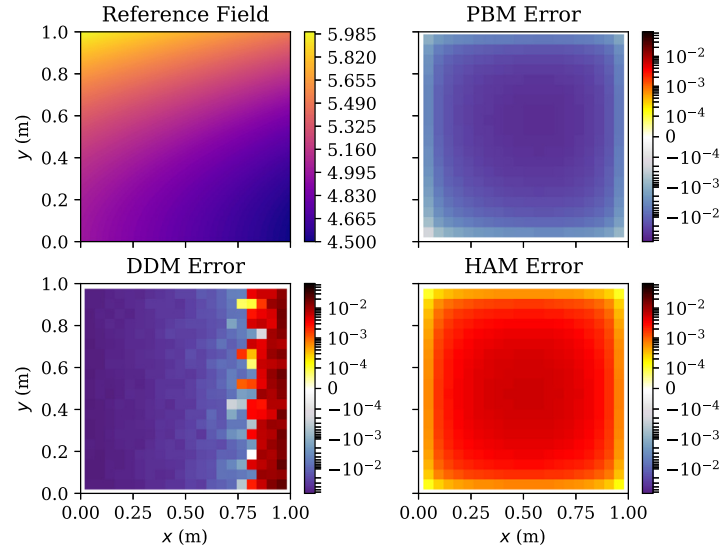


Fig. 23. Solution 2k1, $\alpha = -0.5$: Reference temperature field and relative ℓ_2 -errors of PBM, DDM and HAM at $t = 5$ s.

incorrect modeling of P . However, we do have

$$\Delta t \tilde{\sigma}_P \approx \hat{\sigma} \quad (25)$$

under the assumption that incorrect modeling of P dominates all other sources of error in the PBM (including discretization error).

In Fig. 27, we illustrate that the approximation (25) is valid for Solution 2P2 with $\alpha \in \{0.7, 1.5\}$. The right-hand side of the figure illustrates the true corrective source term $\hat{\sigma}^{n+1}$, as defined in Eq. (16), at the time $t = 0.1$ s. The figure's left-hand side illustrates $\Delta t \tilde{\sigma}_P = \Delta t \epsilon_P / (\rho c_V)$, which is equal to $\Delta t P$ for the choices of \tilde{P} , ρ and c_V used in the present work. As can be seen from the figure, the top and bottom pairs are visually indistinguishable.¹² This indicates that Eq. (25) is a reasonable approximation when it

is known that a PBM suffers from significantly incorrect modeling of P .

We now consider the case where we model a system with unknown conductivity k . We write $k = \tilde{k} + \epsilon_k$, where \tilde{k} is our estimate of the system's conductivity ($\tilde{k} = 1$ for the numerical experiments considered in the present work), and ϵ_k is the error of the estimate. Inserting into the 2D version of Eq. (8) yields

$$\begin{aligned} \int_V \rho c_V \frac{\partial T}{\partial t} dV &= \left(kA \frac{\partial T}{\partial x} \right)_e - \left(kA \frac{\partial T}{\partial x} \right)_w \\ &\quad + \left(kA \frac{\partial T}{\partial y} \right)_n - \left(kA \frac{\partial T}{\partial y} \right)_s + \int_V P dV \\ &= \left((\tilde{k} + \epsilon_k)A \frac{\partial T}{\partial x} \right)_e - \left((\tilde{k} + \epsilon_k)A \frac{\partial T}{\partial x} \right)_w \\ &\quad + \left((\tilde{k} + \epsilon_k)A \frac{\partial T}{\partial y} \right)_n \end{aligned}$$

¹² With the obvious exception that $\hat{\sigma}^{n+1}$ is discrete while we have shown $\Delta t \tilde{\sigma}_P$ as a continuous field since P , and thus also $\hat{\sigma}_P$, is known analytically everywhere.

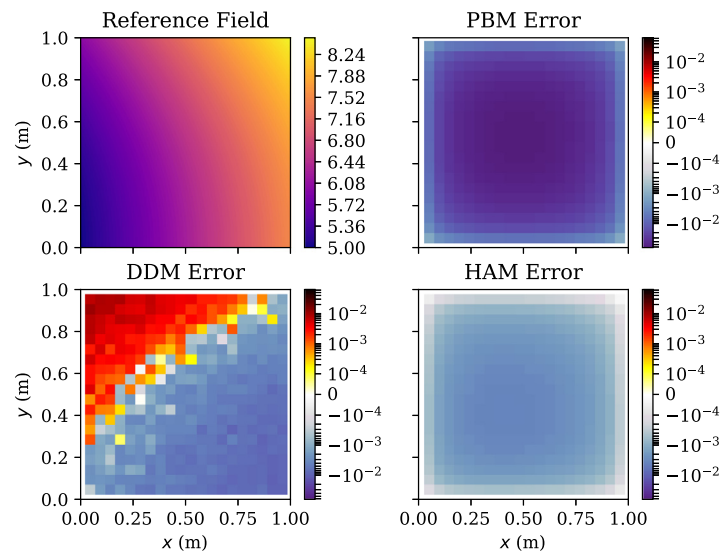


Fig. 24. Solution 2k1, $\alpha = 2.5$: Reference temperature field and relative ℓ_2 -errors of PBM, DDM and HAM at $t = 5$ s.

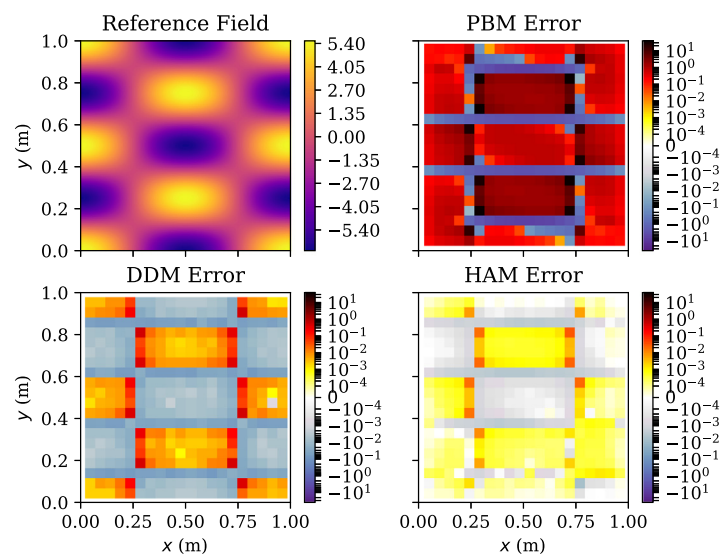


Fig. 25. Solution 2k2, $\alpha = -0.5$: Reference temperature field and relative ℓ_2 -errors of PBM, DDM and HAM at $t = 5$ s.

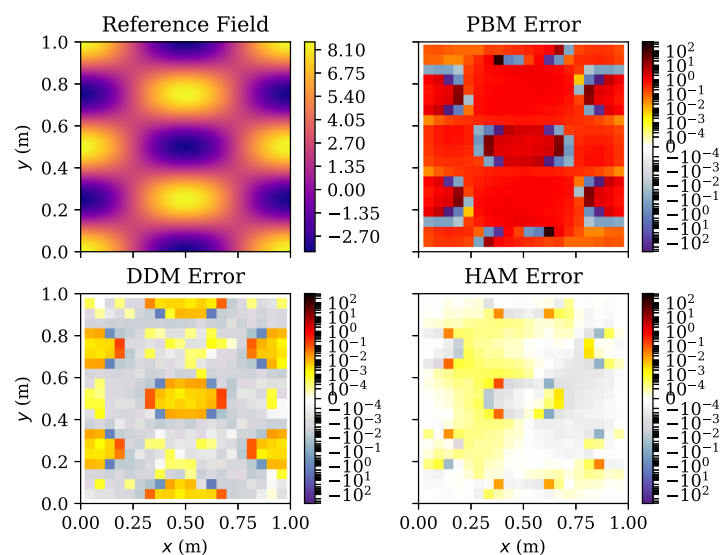


Fig. 26. Solution 2k2, $\alpha = 2.5$: Reference temperature field and relative ℓ_2 -errors of PBM, DDM and HAM at $t = 5$ s.

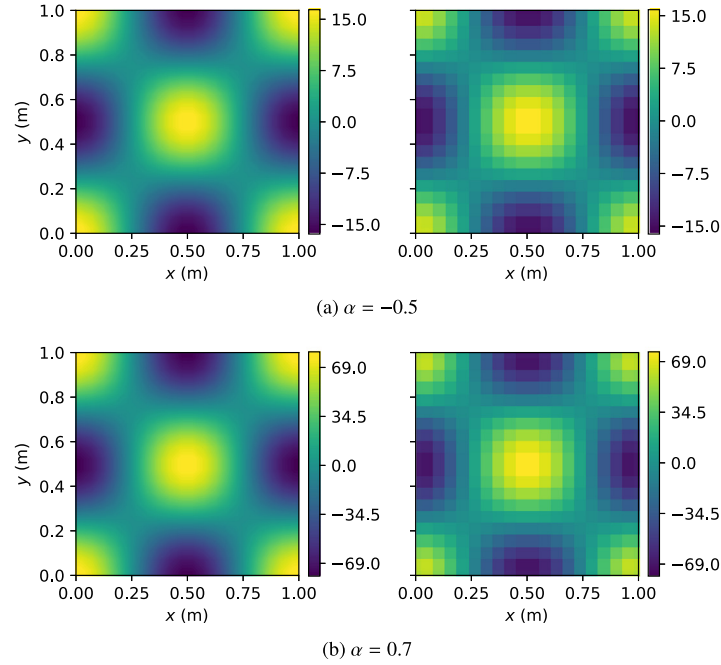


Fig. 27. Comparison of $\hat{\sigma}$ (left) and $\Delta t \hat{\sigma}_k$ (right) for Solution 2P2 at $t = 0.1$ s.

$$\begin{aligned}
 & - \left((\tilde{k} + \epsilon_k) A \frac{\partial T}{\partial y} \right)_s + \int_V P \, dV \\
 & = \left(\tilde{k} A \frac{\partial T}{\partial x} \right)_e - \left(\tilde{k} A \frac{\partial T}{\partial x} \right)_w \\
 & + \left(\tilde{k} A \frac{\partial T}{\partial y} \right)_n - \left(\tilde{k} A \frac{\partial T}{\partial y} \right)_s + \int_V P \, dV \\
 & + \left(\epsilon_k A \frac{\partial T}{\partial x} \right)_e - \left(\epsilon_k A \frac{\partial T}{\partial x} \right)_w + \left(\epsilon_k A \frac{\partial T}{\partial y} \right)_n - \left(\epsilon_k A \frac{\partial T}{\partial y} \right)_s
 \end{aligned}$$

For ease of notation, we define

$$\xi = \left(\epsilon_k A \frac{\partial T}{\partial x} \right)_e - \left(\epsilon_k A \frac{\partial T}{\partial x} \right)_w + \left(\epsilon_k A \frac{\partial T}{\partial y} \right)_n - \left(\epsilon_k A \frac{\partial T}{\partial y} \right)_s, \quad (26)$$

such that we get

$$\begin{aligned}
 \int_V \rho c_V \frac{\partial T}{\partial t} \, dV & = \left(\tilde{k} A \frac{\partial T}{\partial x} \right)_e - \left(\tilde{k} A \frac{\partial T}{\partial x} \right)_w \\
 & + \left(\tilde{k} A \frac{\partial T}{\partial y} \right)_n - \left(\tilde{k} A \frac{\partial T}{\partial y} \right)_s \\
 & + \int_V P \, dV + \xi.
 \end{aligned} \quad (27)$$

To enable a comparison with Eq. (16), we need to discretize the equation above. To this end, we discretize the spatial domain into a grid of $N_j \times N_i$ grid cells, where integral and half-integral indices are used to denote quantities evaluated at cell centers and cell faces, respectively. As such, ξ for a grid cell centered at (x_j, y_i) is

$$\begin{aligned}
 \xi_{j,i} & = \left(\epsilon_k A \frac{\partial T}{\partial x} \right)_{j+1/2,i} - \left(\epsilon_k A \frac{\partial T}{\partial x} \right)_{j-1/2,i} \\
 & + \left(\epsilon_k A \frac{\partial T}{\partial y} \right)_{j,i+1/2} - \left(\epsilon_k A \frac{\partial T}{\partial y} \right)_{j,i-1/2}.
 \end{aligned}$$

We now make the following approximations

$$\begin{aligned}
 (\epsilon_k)_{j+1/2,i} & \approx (\epsilon_k)_{j,i}, & (\epsilon_k)_{j-1/2,i} & \approx (\epsilon_k)_{j,i}, \\
 (\epsilon_k)_{j,i+1/2} & \approx (\epsilon_k)_{j,i}, & (\epsilon_k)_{j,i-1/2} & \approx (\epsilon_k)_{j,i}
 \end{aligned}$$

$$\begin{aligned}
 \left(\frac{\partial T}{\partial x} \right)_{j+1/2,i} & \approx \frac{T_{j+1,i} - T_{j,i}}{x_{j+1} - x_j}, & \left(\frac{\partial T}{\partial x} \right)_{j-1/2,i} & \approx \frac{T_{j,i} - T_{j-1,i}}{x_j - x_{j-1}} \\
 \left(\frac{\partial T}{\partial y} \right)_{j,i+1/2} & \approx \frac{T_{j,i+1} - T_{j,i}}{y_{i+1} - y_i}, & \left(\frac{\partial T}{\partial y} \right)_{j,i-1/2} & \approx \frac{T_{j,i} - T_{j,i-1}}{y_i - y_{i-1}}
 \end{aligned}$$

which yield

$$\begin{aligned}
 \xi_{j,i} & = (\epsilon_k A)_{j,i} \left(\frac{T_{j+1,i} - T_{j,i}}{x_{j+1} - x_j} - \frac{T_{j,i} - T_{j-1,i}}{x_j - x_{j-1}} \right. \\
 & \quad \left. + \frac{T_{j,i+1} - T_{j,i}}{y_{i+1} - y_i} - \frac{T_{j,i} - T_{j,i-1}}{y_i - y_{i-1}} \right).
 \end{aligned} \quad (28)$$

Using the above discretization for ξ and the standard Implicit Euler FVM discretization for the other terms of Eq. (27), we obtain

$$\mathbb{A} \mathbf{T}^{n+1} = \mathbf{b}(\mathbf{T}^n) + \Delta t \hat{\sigma}_k, \quad (29)$$

with \mathbb{A} and \mathbf{b} as in Eq. (13) and $\hat{\sigma}_k = \xi / (\rho c_V)$. Finally, we observe that we have

$$\Delta t \hat{\sigma}_k \approx \hat{\sigma} \quad (30)$$

when incorrect modeling of k is the dominant source of error in the PBM.

The true corrective source term $\hat{\sigma}$ is compared to $\Delta t \hat{\sigma}_k$ in Fig. 28 for Solution 2k1 with $\alpha = -0.5$ and $\alpha = 0.7$ at time $t = 0.1$ s. From the figure, it is clear that the approximation (30) holds well in the interior of the domain. However, at the domain boundaries, comparatively large discrepancies are visible. There are two main contributions to these discrepancies. The first is the approximations made when discretizing ξ , and the other is the influence of the discretization error in the original PBM on $\hat{\sigma}$. Irrespective of where the discrepancies originate from, their presence illustrates that it is generally advisable to perform interpretations of the corrective source term in the interior of the domain, such as to avoid the influence of boundary effects.

Eqs. (25) and (30) have two main areas of application: model improvement and DNN sanity checks. We first consider model improvement, which can be performed by conducting some kind of regression analysis on the true corrective source term $\hat{\sigma}$ to

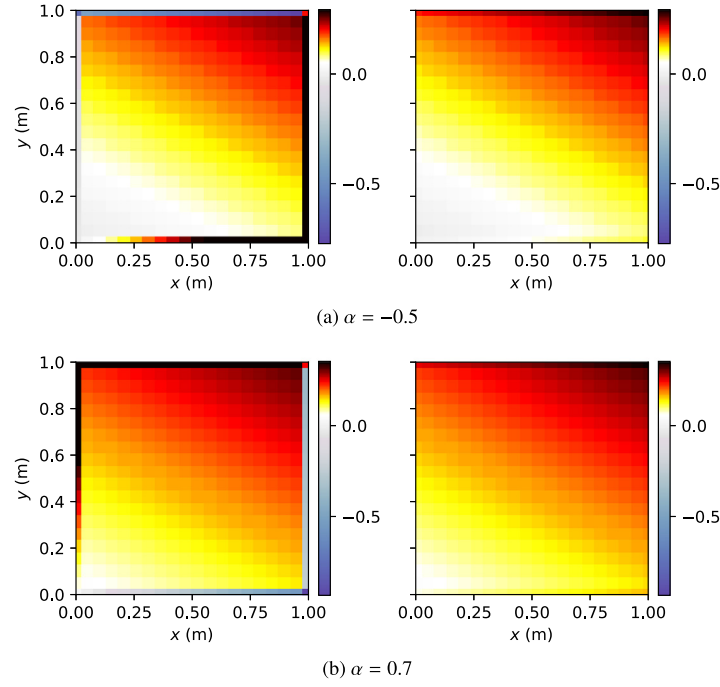


Fig. 28. Comparison of $\hat{\sigma}$ (left) and $\Delta t \hat{\sigma}_k$ (right) for Solution 2k1 at $t = 0.1$ s.

approximate it with some analytic expression. For example, the symbolic regression techniques studied by Vaddireddy et al. [15] are prime candidate for performing this kind of regression analysis. Once a regressed expression for $\hat{\sigma}$ has been identified, this expression can be related to the modeling errors ϵ_P and ϵ_k using the applicable approximation (25) or (30). The modeling of P or k in the PBM-component of the CoSTA model can then be updated according to estimated modeling error. Since CoSTA benefits from improved PBM accuracy, this would increase the accuracy of the CoSTA model.¹³

For sanity checks, we observe first that a successfully trained DNN_{σ} is – by definition – a DNN_{σ} for which we have $\hat{\sigma}_{\text{NN}} \approx \hat{\sigma}$. So let us insert $\hat{\sigma}_{\text{NN}}$ into Eqs. (25) or (30) (depending on which is applicable). For systems whose PBM is dominated by a single error source, we expect the applicable approximation to hold well even with this modification. This allows us to obtain a relation between $\hat{\sigma}_{\text{NN}}$ and ϵ_P or ϵ_k . Of course, for *a priori* predictions, the modeling error ϵ_P or ϵ_k is not known exactly, but it may be known that they are bounded within some range. As a simple example, suppose the we know that a room is heated by some heater whose precise power is unknown. Moreover, suppose that the heater manufacturer specifies that their heaters output $50 \text{ W/m}^3 \pm 10\%$. When modeling the effect of the heater on the room temperature, it is then natural to use $P = 50 \text{ W/m}^3$. Then, ϵ_P should lie within the range $[-5 \text{ W/m}^3, 5 \text{ W/m}^3]$. Eq. (25) then gives us a corresponding bound for $\hat{\sigma}_{\text{NN}}$. This is generally not a hard bound, since $\hat{\sigma}_{\text{NN}}$ is also influenced by any other errors in the PBM, but it gives a ball park estimate for $\hat{\sigma}_{\text{NN}}$. If we suddenly observe that DNN_{σ} produces a $\hat{\sigma}_{\text{NN}}$ well outside the estimated ball park (e.g. a $\hat{\sigma}_{\text{NN}}$ corresponding to negative ϵ_k , in our example), this is a clear sign of DNN misbehavior. As such, physic-based interpretation of $\hat{\sigma}_{\text{NN}}$ using the framework presented in this section provides a valuable sanity-check for the DNN of the CoSTA model.

We conclude this section with two remarks on the applicability of the interpretability approach outlined above. First of all,

if two or more error source contribute significantly to the overall error of a PBM, the approximations underpinning the analyses above do not hold. Also, it is not straightforward to separate the contributions from the different error source. Interpretation of the corrective source term using the suggested approach will then be limited. However, this is a general problem with inverse solution methods, so the issue is not unique to CoSTA. Secondly, we highlight that, when a single error source is dominating, the approach above is not limited to capturing the effects of heat generation or conductivity in the heat equation. To the contrary, analogous calculations can equally well be carried out for other governing equations with other parameters.

5.4. The impact of noise

Just like other modeling approaches utilizing data-driven techniques, CoSTA is potentially vulnerable to noise. Indeed, if the reference data (T_{ref} in our case) contains noise, this noise will be embedded in the corrective source term $\hat{\sigma}$ from Eq. (5). It is therefore beneficial to remove as much noise as possible from the reference data before it is used to train the DNN that is part of the CoSTA model. In particular, it is advised to minimize high-frequency noise, since this noise will have the greatest impact on the magnitude of the corrective source term. If the nature of noise is statistically similar across the snapshots in time, conducting a principal component analysis for denoising the data could be helpful. Still, it is not necessarily so that all noise must be removed for CoSTA to be usable. If the noise is truly random, proper regularization of the DNN should ensure that $\hat{\sigma}_{\text{NN}}$ is not detrimentally affected by the noise. Using the DNN in a CoSTA model does not restrict the choice of regularization techniques, so standard regularization techniques like dropout, weight regularization and early stopping are all applicable.

6. Conclusion

In this work, we presented the Corrective Source Term Approach (CoSTA) to Hybrid Analysis and Modeling (HAM). The method exploits the universal approximation properties of a deep

¹³ The regression analysis and PBM update can be performed before the DNN of the CoSTA model is trained, such that the proposed improvement scheme does not incur any extra DNN training cost.

neural network (DNN) to generate a correction term that compensates for the unmodeled/unresolved physics in a physics-based model (PBM). In a series of numerical experiments on two-dimensional heat diffusion problems, we compared the performance of the CoSTA-based HAM to PBM and the data-driven model (DDM). The two major conclusions from the study are as follows:

- In terms of **predictive accuracy**, CoSTA for two dimensional heat diffusion problems involving unknown physics is **several orders of magnitude more accurate** than comparable PBM and DDM both for interpolation as well as **extrapolation** cases.
- It is also demonstrated that the CoSTA-generated corrective source term can be **subjected to physical interpretation** leading to a **better understanding of the underlying physics**. In fact, **physical laws** (like the conservation of energy) can be used to put a **sanity check** on the predictions of the **DDM-part** of CoSTA models. Such sanity checks are foreseen to result in more reliable models, resulting in increased penetration of DDM in high-stakes applications.

Despite the demonstrated strengths of CoSTA, there are still some areas for improvement. For example, the DDM used does not consider all known information of the scenario in which the model is applied. More specifically, we do not inform the DDM of the parameter α , even if it might be reasonable to assume that α is known. One of the reasons for not exploiting the knowledge about the parameter's value is that the vanilla neural network architecture used in this work downweights the importance of the parameter if it is fed at the input layer. Based on our recent research work in [17–19], it could be desirable to **inject this knowledge in an intermediate hidden layer leading to smoother and more certain solutions**. Another knowledge that is not yet exploited is the temporal correlation of the time series. This can be addressed through the use of long short-term memory (LSTM) network. A potential future extension of the work is proposed in these two directions. Lastly, as explained earlier, we chose to demonstrate the effectiveness of the proposed approach using synthetic data which was devoid of any noise. Since we claim that the approach can be used in the context of predictive digital twins, it would be valuable to apply the approach on real temperature data collected using e.g. high-resolution thermal cameras.

CRedit authorship contribution statement

Sindre Stenen Blakseth: Conceptualization, Methodology, Software, Validation, Formal analysis, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Adil Rasheed:** Conceptualization, Writing – review & editing, Visualization, Supervision, Funding acquisition. **Trond Kvamsdal:** Conceptualization, Writing – review & editing, Supervision, Funding acquisition. **Omer San:** Conceptualization, Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The second and third author are grateful for the support received by the Research Council of Norway and the industrial partners of the following projects: EXAIGON–*Explainable AI systems for gradual industry adoption* (grant no. 304843), *Hole cleaning monitoring in drilling with distributed sensors and hybrid methods* (grant no. 308823), and RaPiD–*Reciprocal Physics and Data-driven models* (grant no. 313909). The fourth author gratefully acknowledges the Early Career Research Program (ECRP) support of the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Award Number DE-SC0019290.

References

- [1] A. Rasheed, O. San, T. Kvamsdal, Digital twin: Values, challenges and enablers from a modeling perspective, *IEEE Access* 8 (2020) 21980–22012, <http://dx.doi.org/10.1109/ACCESS.2020.2970143>.
- [2] O. San, A. Rasheed, T. Kvamsdal, Hybrid analysis and modeling, eclecticism, and multifidelity computing toward digital twin revolution, *GAMM-Mitt.* 44 (2) (2021) e202100007, <http://dx.doi.org/10.1002/gamm.202100007>.
- [3] B. Amos, J.Z. Kolter, OptNet: Differentiable optimization as a layer in neural networks, in: *International Conference on Machine Learning*, PMLR, 2017, pp. 136–145, URL: <https://proceedings.mlr.press/v70/amos17a.html>.
- [4] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, J.Z. Kolter, End-to-end differentiable physics for learning and control, in: *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2018, URL: <https://papers.nips.cc/paper/2018/hash/842424a1d0595b76ec4fa03c46e8d755-Abstract.html>.
- [5] A. Quarteroni, G. Rozza, *Reduced Order Methods for Modeling and Computational Reduction*, Vol. 9, Springer, New York, 2014.
- [6] L. Xiang, C.W. Lee, O. Zikanov, C.-C. Hsu, Efficient reduced order model for heat transfer in a battery pack of an electric vehicle, *Appl. Therm. Eng.* 201 (2022) 117641, <http://dx.doi.org/10.1016/j.applthermaleng.2021.117641>.
- [7] S. Georgaka, G. Stabile, K. Star, G. Rozza, M.J. Bluck, A hybrid reduced order method for modelling turbulent heat transfer problems, *Comput. & Fluids* 208 (2020) 104615, <http://dx.doi.org/10.1016/j.compfluid.2020.104615>.
- [8] T. Li, Y. Gao, D. Han, F. Yang, B. Yu, A novel POD reduced-order model based on EDFM for steady-state and transient heat transfer in fractured geothermal reservoir, *Int. J. Heat Mass Transfer* 146 (2020) 118783, <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2019.118783>.
- [9] S.E. Ahmed, S. Pawar, O. San, A. Rasheed, T. Iliescu, B.R. Noack, On closures for reduced order models—A spectrum of first-principle to machine-learned avenues, *Phys. Fluids* 33 (9) (2021) 091301, <http://dx.doi.org/10.1063/5.0061577>, arXiv:<https://doi.org/10.1063/5.0061577>.
- [10] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707, <http://dx.doi.org/10.1016/j.jcp.2018.10.045>.
- [11] M. Penwarden, S. Zhe, A. Narayan, R.M. Kirby, Multifidelity modeling for physics-informed neural networks (PINNs), *J. Comput. Phys.* (2021) 110844, <http://dx.doi.org/10.1016/j.jcp.2021.110844>.
- [12] Z. He, F. Ni, W. Wang, J. Zhang, A physics-informed deep learning method for solving direct and inverse heat conduction problems of materials, *Mater. Today Commun.* 28 (2021) 102719, <http://dx.doi.org/10.1016/j.mtcomm.2021.102719>.
- [13] J. Bakarji, D.M. Tartakovsky, Data-driven discovery of coarse-grained equations, *J. Comput. Phys.* 434 (2021) 110219, <http://dx.doi.org/10.1016/j.jcp.2021.110219>.
- [14] K. Champion, B. Lusch, J.N. Kutz, S.L. Brunton, Data-driven discovery of coordinates and governing equations, *Proc. Natl. Acad. Sci.* 116 (45) (2019) 22445–22451, <http://dx.doi.org/10.1073/pnas.1906995116>.
- [15] H. Vaddireddy, A. Rasheed, A.E. Staples, O. San, Feature engineering and symbolic regression methods for detecting hidden physics from sparse sensors, *Phys. Fluids Ed. Pick* 32 (2020) 015113, <http://dx.doi.org/10.1063/1.5136351>.
- [16] H. Xu, D. Zhang, N. Wang, Deep-learning based discovery of partial differential equations in integral form from sparse and noisy data, *J. Comput. Phys.* 445 (2021) 110592, <http://dx.doi.org/10.1016/j.jcp.2021.110592>.
- [17] H. Robinson, S. Pawar, A. Rasheed, O. San, Physics guided neural networks for modelling of non-linear dynamics, *Neural Netw.* 152 (2022) 333–345, <http://dx.doi.org/10.1016/j.neunet.2022.07.023>.
- [18] S. Pawar, O. San, B. Aksoylu, A. Rasheed, T. Kvamsdal, Physics guided machine learning using simplified theories, *Phys. Fluids* 33 (1) (2021) 011701, <http://dx.doi.org/10.1063/5.0038929>.

- [19] S. Pawar, O. San, A. N., A. Rasheed, T. Kvamsdal, Model fusion with physics-guided machine learning: projection based reduced order modeling, *Phys. Fluids* 33 (2021) 067123, <http://dx.doi.org/10.1063/5.0053349>.
- [20] R. Rai, C.K. Sahu, Driven by data or derived through physics? A review of hybrid physics guided machine learning techniques with cyber-physical system (CPS) focus, *IEEE Access* 8 (2020) 71050–71073, <http://dx.doi.org/10.1109/ACCESS.2020.2987324>.
- [21] L. von Rueden, S. Mayer, R. Sifa, C. Bauckhage, J. Garcke, Combining machine learning and simulation to a hybrid modelling approach: Current and future directions, in: M.R. Berthold, A. Feelders, G. Kreml (Eds.), *Advances in Intelligent Data Analysis XVIII*, Springer International Publishing, Cham, 2020, pp. 548–560.
- [22] M. Arias Chao, C. Kulkarni, K. Goebel, O. Fink, Fusing physics-based and deep learning models for prognostics, *Reliab. Eng. Syst. Saf.* 217 (2022) 107961, <http://dx.doi.org/10.1016/j.ress.2021.107961>.
- [23] S.S. Blakseth, A. Rasheed, T. Kvamsdal, O. San, Deep neural network enabled corrective source term approach to hybrid analysis and modeling, *Neural Netw.* 146 (2022) 181–199, <http://dx.doi.org/10.1016/j.neunet.2021.11.021>.
- [24] S.S. Blakseth, *Introducing CoSTA: A Deep Neural Network Enabled Approach to Improving Physics-Based Numerical Simulations* (Master's thesis), NTNU, 2021.
- [25] R.J. LeVeque, *Finite-Volume Methods for Hyperbolic Problems*, first ed., Cambridge University Press, 2002.