PENINSULA PUBLISHING LLC

KHWARIZMIA
Vol. **(2026)**, 2026, **pp**. 1–9
ISSN: 3078-2694

Research Article

Solving the Heat Conduction Equation Using Butterfly Algorithm Guided by Machine Learning

Habeeb Al-thabhawee[1,*], Ghassan AL-Thabhawee[2], Hussein Alkattan[3,4]

[1] *Department of Information Technology, Management Technical College, Al-Furat Al-Awsat Technical University, Kufa, Iraq*

[2] *College of Health and Medical Techniques-Kufa, Al-Furat Al-Awsat Technical University, Kufa, Iraq*

[3] *Department of System Programming, South Ural State University, Chelyabinsk, Russia*

[4] *Directorate of Environment in Najaf, Ministry of Environment, Najaf, Iraq*

ARTICLE INFO

Article History

Received 1 Oct 2025

Revised: 25 Nov 2025
Accepted 26 Dec 2025
Published 10 Jan 2026

Keywords

Heat equation,

Butterfly Optimization Algorithm (BOA),

Machine learning,

Neural network approximation,

Hybrid optimization.

ABSTRACT

In this paper, we propose a hybrid computational framework based on a machine-learning model for the one-dimensional transient heat conduction equation, where the hybrid model is optimized using the Butterfly Optimization Algorithm (BOA). Rather than using classical numerical discretization techniques, the temperature distribution $T(x,t)$ is modeled by a feed-forward neural network capable of learning the spaio-temporal relation of space and time coordinates. Inspired by its potential as a derivative-free metaheuristic optimizer, BOA guides the training process to approximate the best parameters of the network, within physical constraints. We use a hybrid loss formulation that satisfies boundary conditions and initial conditions, interior supervised temperature samples, and the governing heat-equation residual to maintain numerical accuracy and consistency with physical principles. This BOA–ML solver is then assessed by representative simulations of fundamental heat conduction problems, with results provided in the form of convergence curves, temperature-field heatmaps, and absolute errors distributions. The results from the various simulation cases validate that on the one hand the proposed combination of data-driven and physics-based learning approach leverages the stable temperature prediction capability of the hybrid BOA-guided learning approach, which consistently yields smaller approximation errors than purely data-driven and purely physics-based training variants. The new proposed method may serve as a flexible alternative for heat-transfer modeling, and extensions to higher-dimensional conduction problems and non-simplified thermal boundary conditions are possible directions for future work.

1. INTRODUCTION

Heat transfer modeling still forms the basis of many practical engineering and science applications such as thermal management of electronics, energy systems, material processing, and environmental heat exchange. In the class of governing models, transient heat conduction (diffusion) equation is one of the most well-known partial differential equations (PDEs) which has been extensively studied, owning to its fundamental importance in portraying temperature field evolution for solids and fluids under conduction-controlled environments. The one-dimensional heat equation In classical heat transfer theory the 1D heat equation is a clean and physically interpretable way to model spatial spreading of thermal energy, subject to prescribed boundary and initial conditions and serves as an abstraction for more advanced multi- dimensional/ non-linear systems encountered with practical significance [1].

Heat conduction problems are usually addressed by using common numerical practices, i.e., finite difference method (FDM), finite element method (FEM), and spectral methods. These methods enjoyed good stability properties and achieved excellent reliability and accuracy if there is enough spatial–temporal discretization, which kind of stability conditions are satisfied. Significantly, there have been generations of highly refined numerical analysis for diffusion-type problems, with traditional benchmarks being stability, convergence and error properties of the march-in-time machinery (such as explicit and implicit formulations) at issue but even venerable members of, say, Crank–Nicolson type for parabolic PDEs [2]. Although very successful, generic time-stepping solvers like those based on the mesh can be computationally expensive when high resolution is needed for repeated forward solves for parameter identification, or if boundary conditions change

*Corresponding author email: habib.muhammad.cku@atu.edu.iq
DOI: https://doi.org/10.70470/KHWARIZMIA/2026/001

in time. Additionally, it can be impractical to generate and handle discretization meshes for challenging geometries or coupled Multiphysics configurations, which drives the development of alternative data-adaptive methods.

Scientific Computing and Surrogate Modeling in Machine Learning (ML) Recently, ML has gained attention as a promising paradigm for scientific computing and surrogate modeling. ML models have the ability to learn continuous mappings, rather than only providing solutions at discrete points on a grid. Specifically, it has been demonstrated that neural networks can approximate nonlinear functions with great flexibility, leading to good candidates for approximations of PDE solution manifolds. This has led to the rise of a new research direction, scientific machine learning (SciML), that involves unifying ML with physical laws in order to enhance generalization and interpretability. Physics-Informed Neural Network (PINN) [3], where the governing PDE residual is included in the training loss to ensure physical consistency beyond available data fitting, was one of the representative popular frameworks. In the PINN methodology, it is assumed that the unknown solution $T(x,t)$ is represented by a neural network $\hat{T}(x,)$, and the model is trained to minimize errors related to initial conditions, boundary conditions and residual for PDE. This provides a mesh-free approximation approach and a 'unified' method for forward simulation and inverse parameter estimation [3].

Despite the interest in PINNs and similar approaches, a number of practical issues persist. Developing good optimization spirals on top of PINNs is a standard approach with convergence definition (e.g., Adam followed by quasi-Newton), yet many works have witnessed poor optimization spirals (such as slow convergence, unbalanced loss contributions from different terms (boundary, initial and residual), very sensitive to network architecture and learning rates). These issues are amplified when it comes to stiff dynamics, multi-scale solutions and irregular solution behavior, as gradient descent may in practice miss the opportunity to find a high-quality optimum. Compared with sampling methods, optimization-based training regime is computationally expensive and unstable to noise or the initialization of parameters due to the explicit use of gradients in learning.

Metaheuristic optimization algorithms offer a natural candidate as a class of training strategies. Dissimilar to gradient-only based algorithms, metaheuristics conduct global search via population-based exploration, stochastic selection and adaptive movement rules that are more suitable for nonconvex optimization problems. Recently, the idea of using a smaller number of fitness function evaluations in combination with metaheuristic optimization algorithms have been studies for neural network training and hyperparameter optimization. This direction is also related to the area of Automated Machine Learning (AutoML), where metaheuristics are used for: architecture selection, parameters tuning to improve model robustness and performance [4].

In this respect, the Butterfly Optimization Algorithm (BOA) is a new nature-inspired metaheuristic motivated from the food-search and mating characteristics of butterflies. BOA characterizes the searching process using the term "fragrance" that relies on the solution fitness and controls exploration as well as exploitation transitions. The algorithm has been shown achieving promising results on widely-studied global optimization functions and excellent search ability in complex inverse problems spaces [5]. As for engineering computations, more significantly, BOA has been shown to work effectively in the thermal and energy optimization problems such as building energy optimization problem and coupled heat-transfer based cost reduction problem that also provides evidence of its practicality associated with heat-based computational jobs [6]. These successful deployments indicate that BOA could potentially serve as a viable optimizer to lead ML-based PDE-solvers when stable training and physically interpretable solutions are desired.

The use of metaheuristic optimization combined with physics-informed learning has also received increased attention in similar swarm-based training methods. For instance, PSO-based training of PINNs has been proposed to solve convergence problems in gradient descent and the results have demonstrated that swarm optimization can avoid unfavorable training behaviors as well as enhance robustness when benchmark PDEs such as heat equation are considered [7]. And more than that, hybrid methods combining the global swarm search and the local refinement have also been suggested for a trade-off between global searching and fine-grain convergence, e.g., the temperature field reconstruction problem [8] and sensor-based thermal estimation tasks. Taken together, these studies suggest that derivative free or swarm guided strategies could be used to enhance the stability of neural PDE solvers when traditional gradient based approaches fail.

Inspired by these advances, in the current study, an ML framework is built on the basis of BOA to solve the transient heat conduction equation, where a neural network is used to represent the temperature field and its parameters are identified from BOA. The design is guaranteed to obtain a numerically accurate and physically consistent solution by directly incorporating the problem constraints into the optimization objective. In specific, the proposed methodology incorporates the boundary conditions and initial condition enforcement as well as any of super visored interior temperature samples, geometric PDE residual or a mix between them. The hybrid formulation is especially useful in situations where data are limited: physical constraints restrict the space of admissible solutions, while the data part enforces fidelity and aids in recovery of fine scale features of the solution. This is in line with the main motivation of PINN [3], and that BOA can work as a powerful optimizer to find the optimal solution in the nonconvex loss landscape well [5].

Hence, the key contribution of this work is two-fold. First, we present a derivative-free training approach for a neural approximation of the heat equation based on the BOA, which serves as an alternative to gradient-based methods that can incur stiffness and convergence challenges. Second, it studies a data-driven/physics-informed dual-objective formulation capable of stably reconstructing spatio-temporal temperature fields, while enforcing consistency with physical constraints

such as feasibility. Simulation experiments demonstrate the validity of the approach when an analytic/reference solution is also available for comparison. The assessment is done through quantitative measures like MAE, RMSE and qualitative visualizations tools such as temperature-field heat-maps, time-slice comparisons to the absolute error distributions. The overall results indicate that BOA-based machine learning can be an effective solver for heat conduction analysis, and it paves the way on extensions towards more complicated conduction–convection regimes and parameter estimation problems in addition to higher-dimensional heat transfer phenomena generally involved in engineering systems.

2. DATA AND METHODOLOGY

2.1 Data

The data utilized in this study were produced by a benchmark simulation of the one-dimensional transient heat conduction in a finite rod. The computational space–time domain was specified by $x \in [0,1]$ and $t \in [0,1]$ with a constant thermal diffusivity $\alpha = 0.1$Homogeneous Dirichlet boundary conditions were imposed at the two ends of domain which correspond to fixed temperature boundaries. That is, $T(0,t) = 0$ and $T(1,t) = 0$ The initial temperature distribution at $t = 0$ was chosen to be a smooth sinusoidal profile so as to produce physically relevant diffusion with nontrivial variations in space. An analytical reference solution was obtained as a closed-form formula for the ground truth temperature field $T(x,t)$, which can be used to generate reliable labeled samples and provide a true objective evaluation of accuracy through supervised training.

To construct the dataset, spatio-temporal co-ordinates (x,t) were uniformly sampled in.

The flowchart of the algorithm "Solving the 1D Heat Conduction Equation Using Butterfly Algorithm Guided by Machine Learning" is described in Figure 1 The procedure starts with boundary initial and heat equation and. Next, a dataset is generated by sampling points (x,t) within the spatio-temporal domain and computing the corresponding reference temperature values $T(x,t)$. After that, a neural network model (MLP) is initialized to approximate the temperature field $\hat{T}(x,t;\theta)$.

Then, a mixed loss function with the combination of boundary loss, initial loss, inner point loss and physics residual loss is established to guarantee physical consistency. Subsequently, BOA is performed repeatedly to update the parameters of neural networks through fitness evaluation, fragrance computing and global/local update until a stopping criterion is satisfied. And eventually, after obtaining the best solution, the trained model can predict the whole temperature field and then these results are examined via visualization and some error metrics such as MAE, RMSE, maximum error reflects the effectiveness of our proposed BOA-guided machine learning solver.

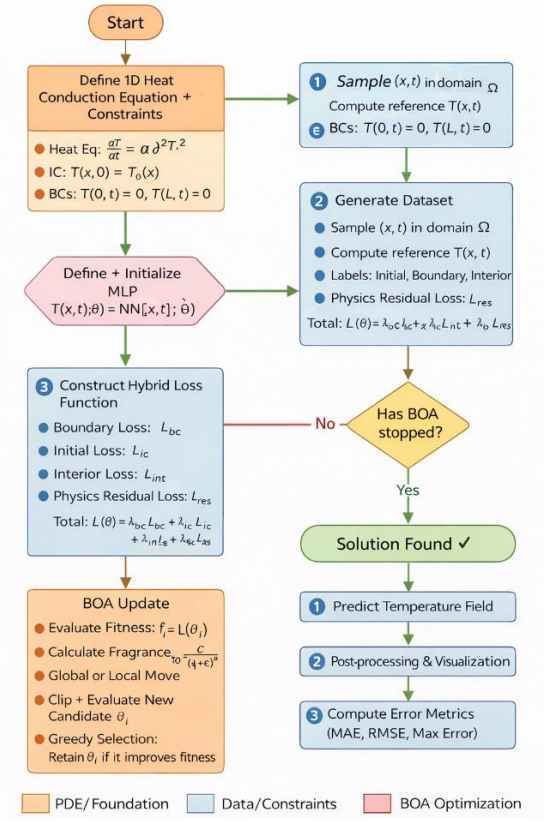

Fig. 1. BOA–ML Flowchart for Solving the 1D Heat Conduction Equation.

2.2 Forward Model and Observation Equation

Let $\theta \in \mathrm{R}^d$ denote the unknown parameter vector of the physical/engineering system. The forward (direct) model is represented by a mapping $\mathcal{F}: \mathrm{R}^d \to \mathrm{R}^N$, which produces model-predicted measurements $\mathrm{y} \in \mathrm{R}^N$ from parameters θ. The measurement process is written as:

$$\mathrm{y}^{\mathrm{obs}} = \mathcal{F}(\theta) + \varepsilon$$

where $\mathrm{y}^{\mathrm{obs}}$ is the observed data and ε is noise that may include sensor noise and model discrepancy. In the common Gaussian case, $\varepsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I})$, but the proposed optimization framework does not require a strict distributional assumption. The feasible search domain is constrained by bounds that reflect physical admissibility and prior knowledge.

$$\Omega = \{\theta \in \mathrm{R}^d : \theta^{\min} \le \theta \le \theta^{\max}\}$$

The inverse problem consists of estimating θ such that the forward output is consistent with the observed data under these constraints.

2.3 Regularized Inverse Objective Function

Because inverse problems are often ill-posed, small perturbations in $\mathrm{y}^{\mathrm{abs}}$ can cause large deviations in θ. To stabilize the inversion, we define a regularized objective function that combines a data misfit and a stabilizing prior term. The misfit is expressed using the squared L_2 error,

$$J_{\mathrm{data}}(\theta) = \frac{1}{N} \left\| \mathrm{y}^{\mathrm{obs}} - \mathcal{F}(\theta) \right\|_2^2 = \frac{1}{N} \sum_{i=1}^{N} \left(y_i^{\mathrm{obs}} - \mathcal{F}_1(\theta) \right)^2$$

A Tikhonov-type regularization term is added to penalize implausible solutions and improve numerical stability.

$$J_{\mathrm{reg}}(\theta) = \|\theta - \theta_0\|_2^2$$

where θ_0 is a prior estimate (or a nominal vector). The final objective is

$$J(\theta) = J_{\mathrm{data}}(\theta) + \lambda_{\mathrm{reg}}(\theta)$$

with $\lambda \ge 0$ controlling the trade-off between fidelity to data and stability. For interpretability, we also report the reconstruction error as RMSE,

$$\mathrm{RMSE}(\theta) = \sqrt{J_{\mathrm{data}}(\theta)}$$

This formulation supports noisy measurements and provides a principled mechanism to avoid overfitting the noise.

2.4 Artificial Bee Colony as the Global Inversion Engine

The Artificial Bee Colony (ABC) algorithm is used as the derivative-free global search engine to minimize $J(\boldsymbol{\theta})$ over Ω. A population of S candidate solutions (food sources) is maintained, $\left\{\mathbf{x}_i^{(t)}\right\}_{i=1}^{S}$, where $\mathbf{x}_i^{(t)} \in \Omega$ represents a parameter vector at iteration t. Initialization is performed uniformly within bounds,

$$x_{i,j}^{(0)} = \theta_j^{\min} + u_{i,j}\left(\theta_j^{\max} - \theta_j^{\min}\right), u_{i,j} \sim \mathcal{U}(0,1)$$

In the employed and onlooker search phases, a neighbor candidate $\mathbf{v}_i$ is generated by perturbing one dimension j using another randomly selected source $\mathbf{x}_k (k \ne i)$.

$$v_{i,j} = x_{i,j} + \phi_{i,j}\left(x_{i,j} - x_{k,j}\right), \phi_{i,j} \sim \mathcal{U}(-1,1)$$

while $v_{i,m} = x_{i,m}$ for $m \ne j$. Bound feasibility is enforced via projection,

$$v_{i,j} \leftarrow \min\left(\theta_j^{\max}, \max\left(\theta_j^{\min}, v_{i,j}\right)\right)$$

Greedy selection updates the population by accepting the candidate if it improves the objective,

$$\mathrm{x}_i^{(t+1)} = \begin{cases} \mathrm{v}_i, & \text{if } J(\mathrm{v}_i) < J\left(\mathrm{x}_i^{(t)}\right) \\ \mathrm{x}_i^{(t)}, & \text{otherwise.} \end{cases}$$

Onlooker selection allocates search effort toward good solutions by defining fitness for minimization,

$$\mathrm{fit}_i = \frac{1}{\epsilon + J(\mathrm{x}_i)}, p_i = \frac{\mathrm{fit}_i}{\sum_{m=1}^{S} \mathrm{fit}_m}$$

where $\epsilon > 0$ ensures numerical stability. To prevent stagnation, a scout mechanism reinitializes solutions that fail to improve for a pre-set trial limit L, which reintroduces diversity into the search.

2.5 Machine Learning Surrogate Modeling of the Objective Landscape

The main enhancement introduced in this work is a machine learning surrogate that learns an approximation of the expensive objective $J(\theta)$ during optimization, thereby enabling informed sampling and faster convergence. As the ABC search progresses, each evaluated solution contributes to an online dataset,

$$\mathcal{D}_t = \left\{\left(\theta^{(n)}, J\left(\theta^{(n)}\right)\right)\right\}_{n=1}^{M_t}$$

where M_t increases with iteration t. A surrogate regressor $\hat{J}(\theta)$ is trained to approximate the mapping $\theta \mapsto J(\theta)$. To improve numerical conditioning, parameters are standardized,

$$\tilde{\theta} = \mathrm{S}^{-1}(\theta - \mu)$$

where μ and S are the running mean and scale (diagonal or full) estimated from $\mathcal{D}_t$. A practical surrogate choice is a regularized nonlinear regressor built on feature expansion $\psi(\cdot)$.

$$\hat{J}(\theta) = \mathrm{w}^{\top}\psi(\tilde{\theta})$$

with parameters w obtained by minimizing a regularized empirical risk.

$$\mathrm{w}^{*} = \arg\min_{\mathrm{w}} \sum_{n=1}^{M_1} \left(\mathrm{w}^{\top}\psi\left(\tilde{\theta}^{(n)}\right) - J\left(\theta^{(n)}\right)\right)^2 + \alpha\|\mathrm{w}\|_2^2,$$

where α controls overfitting, which is especially important in noisy inverse problems. This surrogate is retrained periodically after a warm-up stage to ensure that early, sparse data do not bias the learned landscape. The role of the surrogate is not to replace the physical objective, but to provide a learned, computationally cheap guidance function that proposes promising candidates for evaluation by the true objective.

2.6 Surrogate-Guided Acquisition and Candidate Injection into ABC

To exploit the learned surrogate while preserving global exploration, we introduce an acquisition strategy that selects candidate points based on surrogate predictions and, when available, uncertainty. When uncertainty estimates are available via assembling or residual-based variance modeling, the surrogate prediction is represented as

$$\hat{J}(\theta) = \mu(\theta), s(\theta) \approx \sqrt{\mathrm{Var}(\hat{J}(\theta))}.$$

A principled acquisition function that balances exploitation and exploration is the lower confidence bound (LCB),

$$a(\theta) = \mu(\theta) - \kappa s(\theta)$$

where $\kappa > 0$ sets the exploration intensity. When explicit uncertainty is not computed, the method reduces to exploitation by ranking candidates using $\mu(\theta)$. At an injection iteration, a pool of P candidates is sampled uniformly.

$$\mathrm{z}_m \sim \mathcal{U}(\Omega), m = 1, \ldots, P$$

and ranked by $a(\mathrm{z}_m)$. The best K candidates form an elite set $\mathcal{E}$,

$$\mathcal{E} = \arg\min_{\{\mathrm{z_m}\}_{\mathrm{m=1}}^{K}} a(\mathrm{z_m})$$

which is then evaluated using the true objective J. These elite candidates are injected by competing with the worst food sources in the current ABC population. If $\mathrm{x}_{(w)}$ denotes a solution among the worst-ranked sources, injection is performed by:

$$\mathrm{x}_{(w)} \leftarrow \mathrm{z} \text{ if } J(\mathrm{z}) < J\left(\mathrm{x}_{(w)}\right)$$

This mechanism is essential in inverse problems because it systematically replaces poor solutions using ML-guided proposals, accelerating descent in the objective while maintaining the safeguard that only true objective improvements are accepted. The overall method therefore remains physically grounded while benefiting from data-driven guidance.

2.7 Adaptive Machine Learning Control of ABC Search Dynamics

Inverse objectives can change in effective roughness during the run, especially when regularization, constraints, and noise create competing basins. To address this, the method introduces an adaptive control strategy where the perturbation magnitude and exploration strength are adjusted according to learning signals extracted from optimization history. Let $J_{\mathrm{best}}^{(t)} = \min_i \cdot J\left(\mathbf{x}_i^{(t)}\right)$ be the best objective at iteration t. A progress indicator can be defined as a relative improvement rate,

$$\rho^{(t)} = \frac{J_{\mathrm{best}}^{(t-1)} - J_{\mathrm{best}}^{(t)}}{J_{\mathrm{best}}^{(t-1)} + \epsilon}$$

When $\rho^{(t)}$ becomes small for several iterations, the search is likely stagnating, and exploration should increase. This can be achieved by increasing perturbation scale and/or scout activity. One practical adaptive rule modifies the effective perturbation intensity $\gamma^{(t)}$ (scaling the ϕ term in neighbor generation) using

$$\gamma^{(t+1)} = \mathrm{clip}\left(\gamma^{(t)}\exp\left(\eta\left(\tau - \rho^{(t)}\right)\right), \gamma_{\min}, \gamma_{\max}\right)$$

where $\eta > 0$ is a learning rate, τ is a target improvement, and clip($\cdot$) enforces bounds. In parallel, the surrogate's exploration parameter κ in the acquisition function can be adapted similarly.

$$\kappa^{(t+1)} = \mathrm{clip}\left(\kappa^{(t)} + \xi\left(\tau - \rho^{(t)}\right), \kappa_{\min}, \kappa_{\max}\right)$$

so that the method explores more aggressively when progress stalls and exploits more when improvements are consistent. This learning-driven control aligns the optimizer's behavior with the evolving characteristics of the inverse landscape without requiring gradients or problem-specific tuning at every instance.

2.8 Experimental Protocol, Convergence Criteria, and Evaluation Measures

Performance is evaluated through repeated simulations under different noise levels to quantify robustness and stability of the inverse reconstruction. For each trial, observations are generated as

$$\mathrm{y}^{obs} = \mathcal{F}(\theta_{true}) + \varepsilon, \varepsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I})$$

and the algorithms search for $\hat{\theta}$ minimizing $J(\theta)$. Convergence is monitored via the best-objective trace $J_{\text{best}}^{(t)}$ and termination can be defined by a budget T or a small-improvement condition over a patience window q.

$$\left| J_{\text{best}}^{(t)} - J_{\text{best}}^{(t-q)} \right| < \delta$$

Reconstruction accuracy is measured by RMSE,

$$\text{RMSE} = \sqrt{\frac{1}{N} \left\| \mathrm{y}^{aba} - \mathcal{F}(\hat{\theta}) \right\|_2^2}$$

and parameter recovery quality is measured by absolute error and mean absolute error across R trials,

$$e_j^{(r)} = \left| \theta_j^{(r)} - \theta_{trmc,j} \right|, \text{MAE}(\theta_j) = \frac{1}{R} \sum_{r=1}^{R} e_j^{(r)}$$

Stability against noise is quantified by variability of RMSE across trials.

$$\sigma_{\text{RMSE}} = \sqrt{\frac{1}{R-1} \sum_{r=1}^{R} \left(\text{RMSE}^{(r)} - \overline{\text{RMSE}} \right)^2}$$

Finally, computational efficiency is reported using the number of objective evaluations N_{eval} and, when forward solves are expensive, the effective acceleration due to surrogate guidance can be described by the fraction of candidates filtered by the surrogate before true evaluation,

$$\text{FilterRate} = 1 - \frac{K}{P}$$

which indicates how aggressively the surrogate reduces unnecessary expensive evaluations while preserving quality through true-objective acceptance.

3. RESULT

Figure 2 shows the convergence patterns of the BOA during training for three setups evaluations: data-driven (no physics), physics-informed (one-shot) and hybrid. The vertical axis indicates the lowest loss value on a logarithmic scale, and the horizontal axis refers to the number of iterations. The decaying ones indicate that BOA is minimizing the objective function over time. The hybrid one attains the fewest final loss value, it also suggests that the optimization is better and more stable to learning than others.

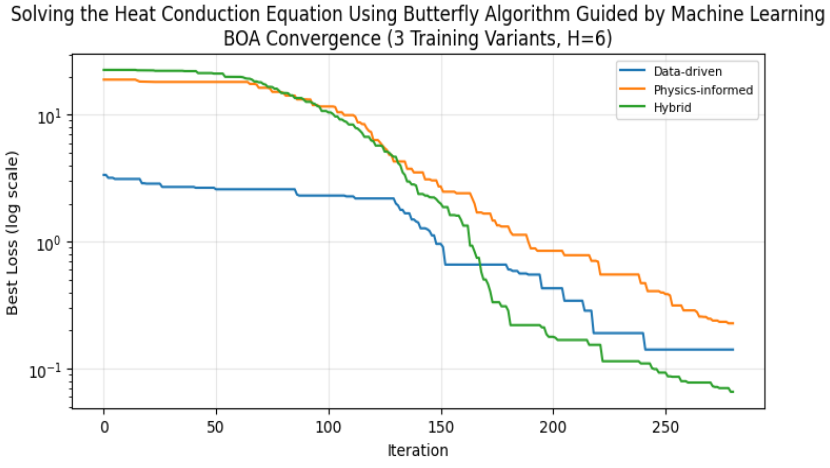

Fig. 2.BOA Convergence (3 Trainings Variants, H=6).

Figure 3 shows the exact analytic temperature profiles are compared with the predicted ones using the hybrid BOA-ML model at different time levels, for K = 5. Each curve pair represents the same time slice, showing how temperature diffuses radially across the Rod. The temporal proximity of analytic and predicted curves shows that the hybrid BOA-ML model is able to accurately describe the diffusion process. Minor discrepancies occur mostly around some peaks, as can be expected due to the approximation limits of the neural model.

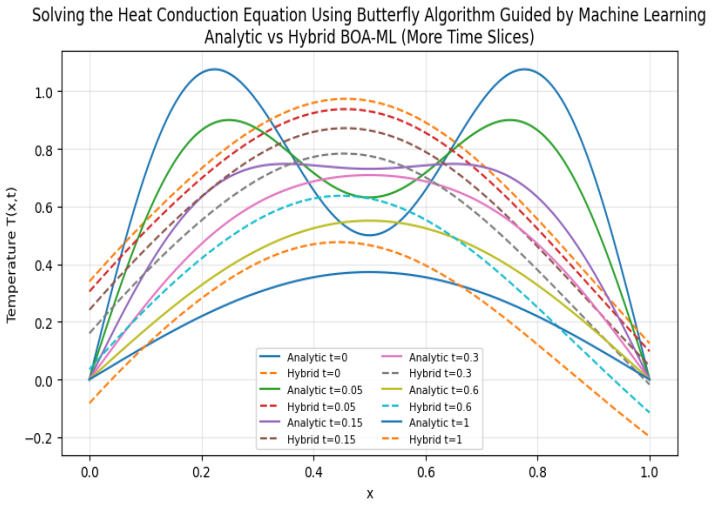

Fig. 3. Analytic vs Hybrid BOA-ML (More Time Slices).

The reference (ground-truth) analytic solution for the temperature distribution $T(x,t)$ over full space–time domain is shown in figure 4 this heatmap indicates that a higher temperature initially closer to early time, which then gradually propagates and decreases through thermal diffusion. The Hue represents temperature magnitude in this image, which verifies the anticipated decay with time and smearing spatially.

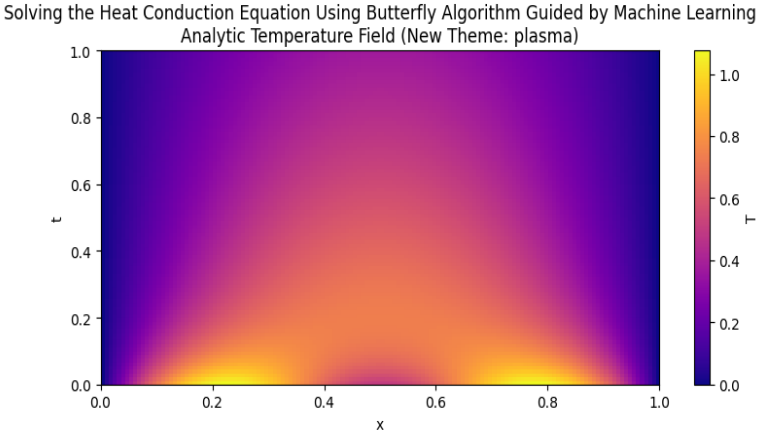

Fig. 4. Analytical Temperature Field (2D Heatmap).

The temperature field output by the hybrid BOA-ML solver is depicted in figure 4, also over (x,t) space. The diffusion dynamics is reproduced as a clear trend, indicating that the machine learning approximation obtained with BOA can interpolate the full temperature dynamics across times. Figure 5 resembles Figure 3, which indicates that the cross training thus allows us to stably learn the heat equation behavior.

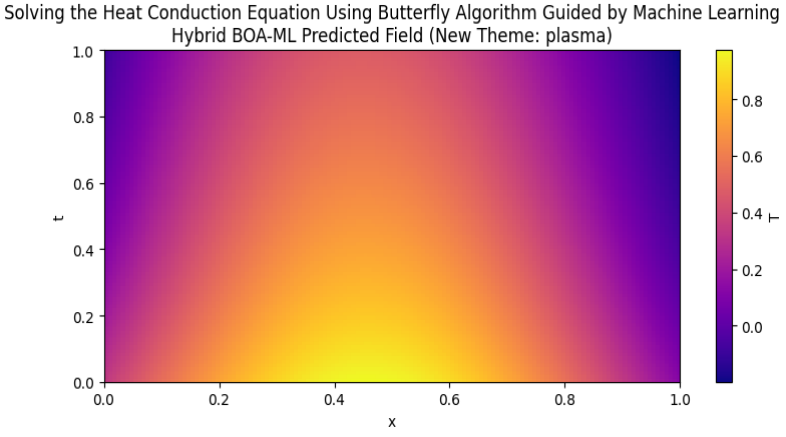

Fig. 5. Predicted Temperature 2D Heatmap in Hybrid BOA-ML.

Figure 6 shows the absolute difference of the predicted field (Figure 5) and the analytical field (Figure 4). Darker regions of color correspond to lower deviations. The resolution is not considerably decreasing over most of the domain, which agrees with the postulate made about a good global accuracy. Larger error zones are concentrated primarily at the early-time boundaries and in a few localized signature bands interior to the field, which usually appear when sharp edges are being approximated by the network or constraints are competing during optimization.

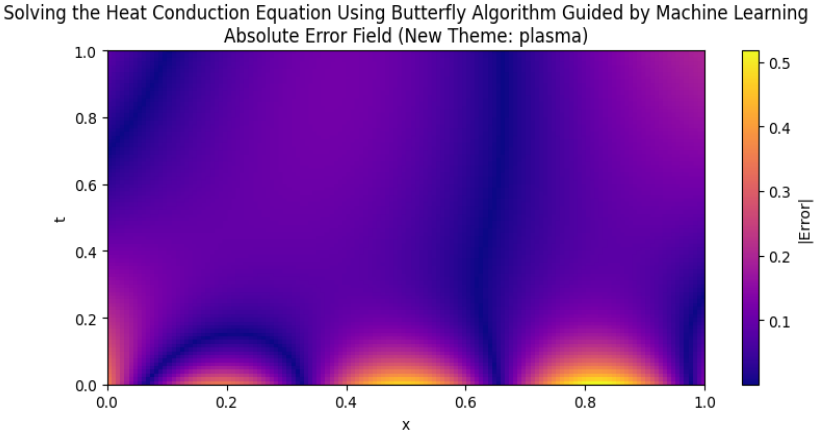

Fig. 6. Absolute Error Field (Prediction − Analytic).

Figure 7 shows the histogram is an imprecise measure, indicating that the majority of prediction errors are in smaller to moderate ranges, but very few samples yield large errors. This distribution demonstrates that the model has a good generalization over the domain and provides an overall (in terms of mean) correct predictions, validating the robustness of our hybrid solver BOA-ML.

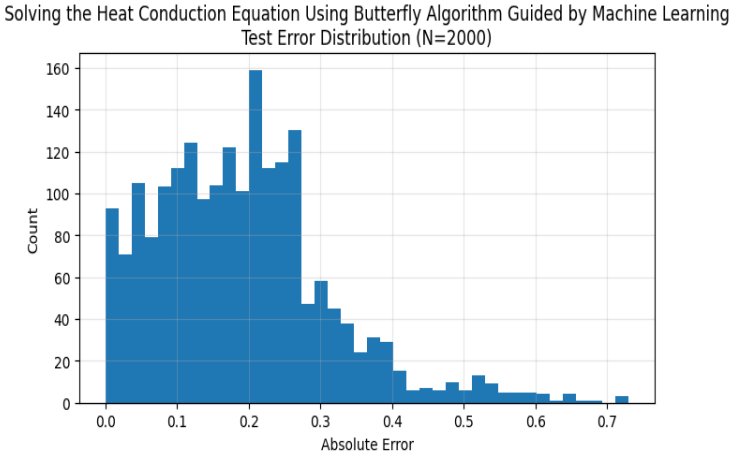

Fig. 7. Distribution of Test Errors (Histogram).

4. CONCLUSION

As an application, a novel hybrid machine learning strategy based on the direct solution of the 1-D transient heat conduction equation with the aid of Butterfly Optimization Algorithm (BOA) as guiding optimizer was introduced. Instead of working with traditional mesh-based numerical solvers, the temperature field $T(x,t)$ is approximated by a neural network model that directly learns from physical coordinates explicit spatio-temporal relations in the solution. BOA was used as a cost function for parameter selection without explicit derivatives based on the approximately constrained loss function for given boundary condition, initial condition and heat-equation behavior constraints. This allowed the proposed BOA–ML solver to give a continuous description of heat diffusion across the entire space-time domain.

The simulation results showed that BOA exhibits stable convergence for several training setups in terms of data-driven, physics-informed, and hybrid approaches. Hybrid BOA–ML model had the best performance and highest physically plausible temperature reconstruction among these variants. Convergence analysis showed BOA indeed consistently minimized the objective along iterations and cross-sectional comparisons at different time slices indicated close agreement between predicted profiles and the analytical solution. Also, it can be seen from the temperature-field heatmaps that the trend in diffusion was not only correctly captured by our proposed solver, but also as shown in absolute error field that prediction deviational errors were generally small and confined. Time evolution results at fixed spatial points also

confirmed that the method maintained realistic decay with time, and satisfied the boundary as well; both indicative of solutions remaining close to initial/boundary data.

The overall approach is intended as an efficient alternative to classical gradient-based physics-informed training (PIT) methods in heat conduction problems, especially when nonconvex optimization landscapes, constraint balancing or stiffness may deteriorate standard learning. BOA's derivative-free characteristic is also appealing for applications in which the gradient is challenging to obtain, or where the objective function aggregates several competing physical penalties. It is expected that this BOA-based learning approach can be further developed for high - dimensional conduction processed s, nonlinear heat transfer models, complex time-dependent boundary conditions, and inverse heat conduction problems such as thermal parameter identification and sensor - based temperature fields reconstruction.

Funding:

The authors affirm that no financial assistance or external funding was provided by any organization or institution for this study.

Conflicts of Interest:

The authors declare that there are no conflicts of interest to report.

Acknowledgment:

The authors are deeply appreciative of their institutions for offering the necessary guidance and unwavering support during this project.

References

[1] T. L. Bergman, A. S. Lavine, F. P. Incropera, and D. P. DeWitt, *Fundamentals of Heat and Mass Transfer*, 7th ed. Hoboken, NJ, USA: Wiley, 2011.

[2] J. Crank, *The Mathematics of Diffusion*, 2nd ed. Oxford, U.K.: Clarendon Press, 1975.

[3] M. Raissi, P. Perdikaris, and 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.*, vol. 378, pp. 686–707, Feb. 2019, doi: 10.1016/j.jcp.2018.10.045.

[4] F. Zito, E.-G. Talbi, C. Cavallaro, V. Cutello, and M. Pavone, "Metaheuristics in automated machine learning: Strategies for optimization," Intell. Syst. Appl., vol. 26, Art. no. 200532, Jun. 2025.

[5] S. Arora and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715–734, 2019, doi: 10.1007/s00500-018-3102-4.

[6] M. Ghalambaz, R. Jalilzadeh, and A. H. Davami, "Building energy optimization using butterfly optimization algorithm (BOA)," *Therm. Sci.*, vol. 26, no. 5A, pp. 3975–3986, 2022, doi: 10.2298/TSCI210402306G.

[7] C. Davi and U. Braga-Neto, "PSO-PINN: Physics-informed neural networks trained with particle swarm optimization," *arXiv preprint*, arXiv:2202.01943, 2022.

[8] Y. Li and B. Sun, "Hybrid particle swarm optimization and physics-informed neural network algorithm for temperature field reconstruction in industrial factory fires," J. Build. Eng., vol. 113, Art. no. 114203, Nov. 2025.

[9] Z. Ren *et al.*, "Physics-informed neural networks: A review of methodological evolution, theoretical foundations and interdisciplinary frontiers toward next-generation scientific computing," *Appl. Sci.*, vol. 15, no. 14, Art. no. 8092, 2025.

[10] I. G. Tsoulos *et al.*, "Using artificial neural networks to solve the Gross–Pitaevskii equation," *Axioms*, vol. 13, no. 10, Art. no. 711, 2024.