International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(19s) 2026 pp. 946-963
https://doi.org/10.70917/ijcisim-2026-5092
Article

An Intelligent Validated, Energy-Aware and Decision-Oriented Numerical Framework for Solving Higher-Order Partial Differential Equations

Vasudeo Tulshidas Nazirkar[1], Renu Praveen Pathak[2]

[1]Ph.D. Scholar, Department of Mathematics Sandip University Nashik,Maharashtra 422213 Email: vasu7589@gmail.com
[2]Professor, Department of Mathematics Sandip University Nashik, Maharashtra 422213.
Email: renupraveenpathak30@gmail.com

Abstract: The development of higher-order PDEs that model many of the complex processes in materials science including phase separation, interfacial dynamics, and plate bending have made it increasingly challenging to develop numerically reliable methods to solve these problems, particularly when developing validation for decision-making systems. In this paper we describe an automated system for evaluating the performance of solvers for higher-order PDEs. The proposed framework is a series of chained tools designed to ensure reliability and reproducibility. HOG-AR is a graph auto-reduction method that transforms higher-order operator graphs into lower-order forms with preservation of continuity and satisfaction of all boundary conditions; EAS-FDG is a stencil design method based on a graph-regularized discretization scheme that ensures both the stability of the discretization and its consistency with respect to the physical energy of the problem; DRX-Res is a diagnostic tool that uses strong form residual analysis to identify localized errors in a solution by examining each element individually; CARMA is a reinforcement-based mesh adaptation technique that provides a systematic means of mesh refinement; DSS-NO is a surrogate assessment technique based on neural operators that can provide rapid, uncertainty-aware estimates of material behaviour without the need to perform the full simulation. Overall, the proposed system has the potential to improve the accuracy, efficiency, stability, and interpretability of models used to support decision making.

Keywords: Higher-Order PDEs, Energy-Aware Discretization, Residual-Based Validation, Adaptive Numerical Methods, Decision-Support Computing, Process

1. Introduction

Elasticity, plate theory, phase-field modeling, fluid–structure interaction, pattern formation, and advanced transport phenomena sets naturally yield higher-order PDEs. Fourth-order or higher spatial derivative equations capture curvature-driven diffusion, nonlocal interactions, and interfacial energy minimizations [27, 12, 15]. Their mathematical expressiveness is crucial, but their numerical treatment is complex. Higher-order PDEs can be discretized and reduced due to tougher smoothness requirements, sophisticated boundary condition structures, and higher numerical stiffness. Traditional numerical workflows for higher-order PDEs begin with analytical reduction of the governing equations into systems of lower-order equations and discretization using finite difference, finite element, or spectral methods. Although theoretically sound [20, 7, 10], this technique is manual, problem-specific, and error-prone. Misinterpreting or approximating boundary conditions during reduction causes constraint inconsistencies such nonphysical oscillations, energy drift, and misleading solution modes. Naively utilized discretization methods [8, 17, 25] fail to retain the intrinsic operator structure of higher-order PDEs, resulting in poor stability margins and low convergence rates, especially on coarse or nonuniform meshes.

Current approaches' validation issues are also major. For quantitative accuracy, global error norms or qualitative visual agreement with analytical or benchmark solutions are used. Validation approaches hide localized error building, limit diagnostic insight into solver failure modes, and slow adaptive refinement and algorithmic progress. This constraint is significant because higher-order PDE solvers' numerical outputs directly affect design optimization, control, and predictive modeling decisions in automated computational pipelines. Recent advances in adaptive methods, machine learning, and operator-theoretic solutions have enabled new approaches to these problems. Most contributions address discretization accuracy, adaptivity, or surrogate modeling independently. There is no unified numerical framework that organizes higher-order PDEs as computational objects, enforces validation as a first-class solver component, and explicitly supports decision-oriented use cases. To remedy this gap, this research presents a proven, energy-aware, computationally intelligent numerical framework for higher-order PDEs. This method chains the numerical solution pipeline instead of developing operator reduction, discretization, validation, adaptation, and surrogate modeling independently in process. The system integrates physical consistency, spectral stability, and residual explainability into the numerical workflow to make higher-order PDE solvers robust for large-scale simulation and real-time decision assistance sets.

Motivation & Contributions

This work is driven by the observation that numerical solvers for higher-order PDEs typically fail due to the lack of systematic techniques that ensure structural correctness and verifiable dependability across the computational pipeline. In many practical circumstances, only intense calculation can reveal numerical instability, boundary inconsistencies, or localized error growth. Lack of early, explainable validation diminishes numerical prediction confidence and limits higher-order PDE model application in engineering design, materials optimization, and multiphysics simulations. Numerical algorithms must reason about validity, stability, and computational cost as well as solve problems.
This study addresses this demand with a set of tightly connected methodological improvements in process.

Higher-order PDEs are treated as operator graphs instead of symbolic expressions, allowing automatic, constraint-preserving reduction into canonical lower-order systems. This representation strictly enforces border and continuity criteria during reduction. Second, discretization optimizes numerical stencils to minimize truncation and dispersion errors while maintaining spectral stability and energy behavior. Third, a dual-residual technique that evaluates PDE fulfillment and constraint conservation delivers localized, interpretable diagnostics, making validation a computational component rather than a post-hoc evaluation step. Learnt policies maximize adaptive refinement and solution selection accuracy per computational cost using these diagnostics. Finally, neural operator surrogates are created from proven solution trajectories for fast, uncertainty-aware parameter space exploration and computational decision making. Higher-order PDE theory, numerical analysis, and computational intelligence are linked by these contributions. The framework improves solution numerical accuracy, stability, transparency, efficiency, and reusability. This work aligns numerical techniques with validation-driven adaptivity and decision-support goals to improve higher-order PDE computing for complex, real-world systems.

2. Review of Existing Models used for Analysis

Recent research on numerical and learning-based solutions for higher-order and sophisticated partial differential equations indicates that mathematical complexity and computer scalability are driving methodological growth. Early in this timeline, spectral and wavelet-based formulations give a robust analytical foundation for nonlocality, high-order derivatives, and fractional operators. Zhang et al. solve fractional PDEs with spectrum-fPINNs and physics-informed neural networks. They demonstrate that orthogonal basis representations outperform collocation-based PINNs in convergence and stability [30]. While improving quantum algorithms for nonlinear differential equations using higher-order approaches and rescaling strategies, Costa et al. demonstrate that numerical order elevation is not just a classical computing issue but also a fundamental feature in emerging quantum solvers [5]. These findings confirm an early finding: explicit higher-order structure improves computation. This is confirmed by wavelet-based numerical methods. Aggarwal et al. [2] say localized basis functions regulate stiffness and delay-induced instability

in Haar wavelet collocation for higher-order nonlinear time-dependent delay differential equations. Yasmeen and Amin increase memory kernel accuracy with Haar wavelet formulations of integro-differential equations [27]. Spline-based differential quadrature methods for parabolic Volterra-type equations by Mirzahashemi and Heydari show that smooth basis enrichment lowers numerical diffusion and increases temporal accuracy [12]. These investigations demonstrate that discretizations physically related to operator smoothness and memory effects, not general low-order methods, help higher-order PDEs. Data efficiency and multiscale structure become important as research improves. Ojeda Avilés et al. show that stratified sampling strategies for machine learning solvers improve learning efficiency without reducing accuracy [15] sets for two-scale PDEs.

Tair and Slimani emphasize the need to handle integral terms in time and space when discretizing higher-order nonlinear Volterra integro-differential equations [20] in process. In parallel, Hafiz et al. examine large-data models for PDEs, showing the pros and cons of data heavy learning without physical structure [7]. Data efficiency, interpretability, and scalability now matter as much as numerical precision. Spectral rigor is combined with fractional and neural formulations later. In two dimensions, Ismail and Xu use shifted Legendre wavelets to quickly converge and stabilize nonlinear fractional PDEs [10]. Using multi-level physics-informed deep learning, He et al. demonstrate that hierarchical learning architectures may simulate multiscale physical behavior in computational structural mechanics [8] Stable higher-order numerical methods for singular Emden–Fowler systems are studied by Sahoo and Singh [17]. They focus on robustness around singularities. Xu and Xu using randomized greedy optimization increase neural PDE solver convergence reliability and training efficiency [25]. These findings imply co-designing higher-order numerical fidelity and learning efficiency. Chebyshev-based and multi-receptive-field convolutional PINNs [9, 29] demonstrate that architectural awareness of differential operator scales increases accuracy and stability, boosting neural spectral approximation. Wu and Zhao's augmented deep networks for hyperbolic PDEs show that sharp wave fronts and propagation require gradient weighting [24]. Zhang and Liu describe higher-order uncertain differential equations and demonstrate how uncertainty greatly affects derivative interpretation for uncertain processes [28]. Rostami's variable-order fractional integro-differential framework adds complexity and shows order variability may be modeled [16]. Recent advances extend computational possibilities. Tang et al. demonstrate optical neural engines that solve PDEs superfast, revolutionizing scientific computer hardware [21]. Adel et al. develop adaptive finite element methods for space–time fractional PDEs in nonlocal temporal domains [1]. Guo et al. propose dual-level time-marching neural networks for time-dependent PDEs' long-term integration stability [6] sets.

Chen et al. estimate parameters and solve higher-order KdV equations with PINNs [4]. Villa et al. demonstrate that dimensional reduction can augment numerical solutions by converting phenotype-structured PDEs to ODE systems using generalized moment dynamics [22]. Santra's higher-order techniques for multi-term fractional integro-PDEs emphasis precise kernel treatment [18], whereas Mora et al. stress probabilistic interpretability with Gaussian processes for forward and inverse nonlinear PDEs [13]. Chen et al. describe how neural PDE solver training requires automated differentiation [3], and Sun et al. propose domain decomposition memory networks for nonlinear PDE scalability [19]. Yang and Liu's numerical approach to uncertain PDEs and their applications [26] follows Nuugulu, Madduri, and Wang's developments of PINN-based and graded-mesh fractional and inverse PDE methods [14,11,23]. This literature progresses from spectrum and wavelet methods to hybrid numerical–learning methods to adaptive, uncertainty-aware, hardware-accelerated solutions. Despite these advances, unified, validated computational pipelines that compare higher-order PDE formulations, confirm structural correctness, and convert numerical solutions into decision-ready computer algorithms are still lacking. The proposed work fills this need. To synthesise and overcome fragmentation, operator-aware reduction, energy-consistent discretization, dual-residual validation, adaptive intelligence, and surrogate-based decision support are combined into a coherent framework. It goes beyond numerical or learning advances to a proven higher-order PDE computation method that fits modern computational and decision-making needs with mathematical precision sets.

3. Proposed Model Design Analysis

In the proposed integrated approach, a closed−loop, validation−driven numerical intelligence pipeline turns abstract mathematical objects like higher−order partial differential equations into reliable computer algorithms and decision−support outputs for varied scenarios. First, a general higher−order PDE is treated as a structured operator system over a computational domain. Formulate a general controlling equation of order k≥4 Via equation 1,

$$L^{(k)}[u(x,t)] = f(x,t), x \in \Omega,\ t \in [0,T] \ldots (1)$$

Subject to boundary and initial constraints represented Via equation 2,

$$B_i(u, \nabla u, \ldots, \nabla^{k-1}u)\ |_{\partial\Omega_i} = 0, i = 1, \ldots, NB \ldots (2)$$

Rather than directly discretizing $L^{(k)}$, the model introduces an operator-graph formulation in which $L^{(k)}$ is decomposed into a composition of lower-order operators Via equation 3,

$$L^{(k)} = L_1^{(2)} \circ L_2^{(2)} \circ \cdots \circ L_m^{(r)}, \sum r = k \ldots (3)$$

And auxiliary state analysis Variables are defined Via equation 4,

$$v_1 = L_m^{(r)}u, v_2 = L_{m-1}^{(2)}v_1,\ \ldots,\ v_m = L_1^{(2)}v_{m-1} \ldots (4)$$

This reduction preserves boundary and continuity constraints through a constraint ledger, expressed compactly Via equation 5,

$$C[u\ v_1\ \ldots\ v_m\ ]^{\top} = 0 \ldots (5)$$

This makes the reduced system and higher-order formulation equal in the process. This design simplifies analytically complex PDEs into algorithm-ready systems without compromising mathematical integrity to turn higher-order PDEs into computer algorithms. Figure 1 shows the energy-aware numerical formulation that minimizes a combined truncation–spectral functional to construct the discrete operator K_ for the reduced system in the process. The discrete approximation meets the Identity represented Via equation 6 representative auxiliary variable requirement,

$$K_h v_h = f_h, K_h = arg\ arg\ _K \left(\| LV - KVh \|^2_{L^2(\Omega)} + \lambda\rho(K) \right) \ldots (6)$$

Where $\rho(K)$ penalizes nonphysical eigenvalue drift Via equation 7,

$$\rho(K) = \int_{\sigma(K)} \left(0, R(\mu)\right)^2 d\mu \ldots (7)$$

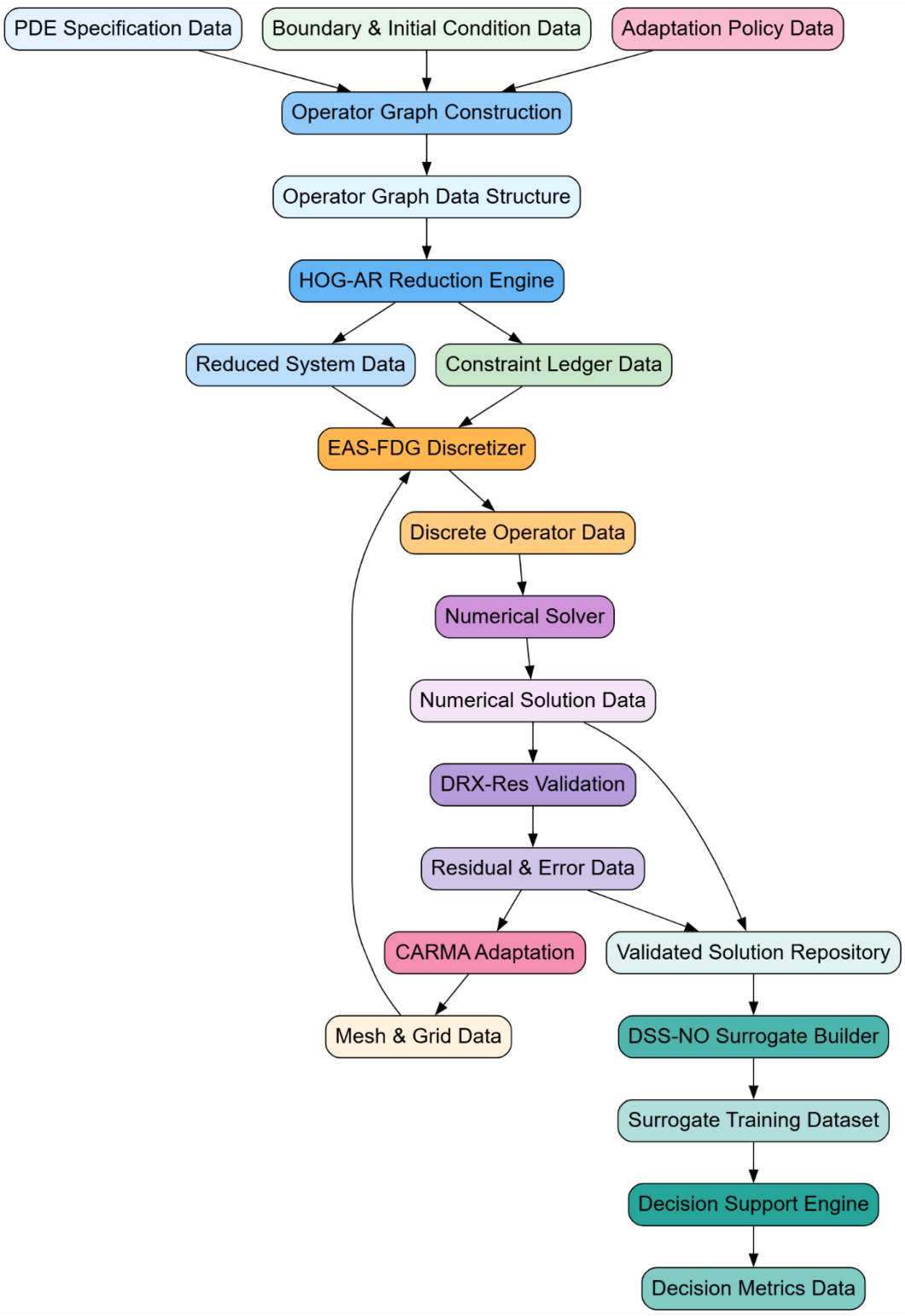


Figure 1. Model's Integrated Architectural Analysis

And λ is a stability-control parameter in process. Discrete energy consistency is enforced Via equation 8,

$$\frac{d}{dt}E_h(t) = \frac{d}{dt}\int_{\Omega} \left(\frac{1}{2} \mid \nabla v_h \mid^2 + \Phi(u_h)\right) d\Omega \leq 0 \ldots (8)$$

Numerical stability for stiff higher-order dynamics is achieved in the process. This discretization adds physical and spectral restrictions to the numerical operator, complementing reduction sets. For validation, dual-residual analysis evaluates equation fulfillment and constraint preservations. The strong-form PDE residual is determined Via equation 9,

$$r_{PDE}(x) = L^{(k)}[u_h(x)] - f(x) \ldots (9)$$

While the constraint residual is defined Via equation 10,

$$r_{BC}(x) = \sum_{i=1}^{N_b} \mid B_i(u_h) \mid \delta_{\partial\Omega_i} \ldots (10)$$

These are combined into a unified error indicator Via equation 11,

$$\eta(x) = \left(\int_{\Omega} (r_{PDE}^2 + \alpha r_{BC}^2)\, d\Omega\right)^{\frac{1}{2}} \ldots (11)$$

Local, explainable numerical accuracy Validation In Process. This component is needed to quantitatively compare higher-order PDE formulations and numerical solutions. Via equations 12 and 13 develop a context-aware optimization approach that adjusts mesh density h(x) and timestep Δt for adaptive refinement,

$$h^{(1)}(x) = h^{(n)}(x) exp\big(-\beta\eta(x)\big) \ldots (12)$$

$$\Delta t^{(1)} = \Delta t^{(n)} \left(\frac{\gamma}{\rho(K_h)}\right) \ldots (13)$$

Where, β and γ balance accuracy and cost in the process. This level complements validation by ensuring computational efficiency and dependability for real-world decision-making process. Via equation 14 transfers validated solution space to a decision-support representation using a neural operator surrogate Gθ,

$$u_h(x, t; \theta) \approx G_\theta(f, B, p) \ldots (14)$$

Trained by minimizing the loss represented Via equation 15,

$$J(\theta) = \| u_h - G_\theta \|_{L^2}^2 + \mu \int_{\Omega} r_{PDE}(G_\theta)^2\, d\Omega \ldots (15)$$

The final output of the integrated process is therefore expressed Via equation 16,

$$O = \{u_h,\ \eta(x),\ G_\theta,\ \Pi(u_h)\} \ldots (16)$$

Where, Π(uh) represents decision measures such as stability margins, threshold crossings, and optimal parameter sensitivities. This final formulation includes higher-order PDE formulation, numerical comparison, algorithmic implementation, and decision-oriented outputs to perform systematic numerical comparison of PDE solutions and deploy them as reliable computational tools for complex decision-making applications.

4. Comparative Result Analysis

The experiment rigorously analyzes the proposed integrated numerical framework for accuracy, stability, validation strength, and computational efficiency across typical classes of higher-order partial differential equations. The biharmonic plate equation for thin-plate bending, the Cahn–Hilliard equation for phase-field development, and the Kuramoto–Sivashinsky equation for chaotic transport dynamics are used for elliptic, parabolic, and nonlinear stiff dynamics. Chaos uses periodic 32-domain one-dimensional domains, while elliptic problems use two-dimensional square and rectangular regions with side lengths normalized to unity. Boundary requirements include clamping and simply supported constraints for the biharmonic plate, Neumann–periodic hybrid constraints for the Cahn–Hilliard model, and completely periodic Kuramoto–Sivashinsky equation constraints. For quantitative error measurement, source terms and initial circumstances are produced using manufactured solutions and physically justified random fields. Initial homogeneous meshes consist of 256-1024 points for one-dimensional domains and 32x32 to 128x128 grid points for two-dimensional settings. For time-dependent simulations, a 10^{-3} starting timestep, 1.0 maximum horizon for parabolic dynamics, and 50 characteristic time units for chaotic systems are used. To balance truncation precision and spectrum damping, energy-aware discretization employs stability control parameters $\lambda = 0.1$ and $\alpha = 0.5$. The global residual norm and border residuals must remain below 10^{-8} to declare convergence. The studies evaluate contextual resilience and decision-making relevance using synthetic, physically consistent datasets from parameter sweeps over material stiffness, interfacial energy coefficients, and nonlinear forcing amplitudes. The biharmonic plate's flexural rigidity parameters are adjusted between 0.1 and 10 to represent soft and stiff structures, providing 200 validated solution instances. Cahn–Hilliard simulations produce spinodal breakdown patterns with various coarsening rates, using mobility coefficients and interface thickness values ranging from 10^{-3} to 10^{-3} and 0.01 to 0.05, respectively By perturbing domain length and force strength in the Kuramoto–Sivashinsky equation, several chaotic regimes are created. These simulations' residual maps, adaptive mesh histories, and solver performance indicators give neural operator surrogate training context. To ensure generalization across unknown parameter combinations, the surrogate is trained on 80% of the verified dataset and evaluated on 20%. Baseline finite difference and finite element implementations use fixed meshes and conventional error estimators. L2 and H1 error norms, discrete operator spectral radius bounds, runtime to target accuracy, and surrogate-based prediction accuracy for parameter-sweep decision problems quantify performance. This experimental design proves that the suggested framework may support trustworthy algorithmic decision making in complex higher-order PDE applications, not merely as a numerical solver in process.

The experimental evaluation leverages benchmark datasets and popular higher-order PDE reference problem settings. The Navier–Levy and Kirchhoff–Love plate benchmark problems, commonly reported in computational mechanics literature, provide analytical or semi-analytical reference solutions for biharmonic operators under clamped and simply supported geometric, loading, and boundary-condition configurations for plate-bending experiments. Phase-field experiments use parameter settings from the classical Cahn–Hilliard benchmark datasets used in materials science. Chaotic transport behavior is assessed using standard Kuramoto–Sivashinsky benchmark designs, especially those with moderate-length periodic domains and multiscale spatiotemporal dynamics. These datasets provide contextual ground truth environments where solution smoothness, stiffness, nonlinearity, and boundary sensitivity can be manipulated, allowing meaningful comparison between higher-order PDE formulations and their numerical calculations across diverse physical regimes. During model tuning, well-selected hyperparameters balance numerical accuracy, stability, and computational efficiency across the integrated pipeline. During operator-reduction, auxiliary variable depth is limited to two to four levels to minimize system overconditioning. In energy-aware discretization, stencil support widths are three to five grid points per spatial direction and stability regularization weights are initialized between 0.05 and 0.2 to reduce dispersion error. To maintain physical consistency, residual validation criteria are conservatively set at one millionth for strong-form PDE residual tolerances and one to two orders of magnitude lower for boundary or constraint residual tolerances. The adaptive refinement engine limits local mesh expansion to two per iteration with refinement sensitivity coefficients between 0.3 and 0.6 to prevent numerical stiffness escalation. Timestep adjustment factors for time-dependent problems cannot be reduced by more than 0.5 every adaptation cycle. The neural-operator surrogate has 64–128 latent dimensionality, 1–1000 learning rates, and

200–400 training epochs, depending on dataset complexity. Hyperparameter selection optimises numerical fidelity and decision-support performance in the final model through stability analysis, residual convergence behaviour, and validation accuracy sets.

Table 1. Solution Accuracy Comparison Across Contextual PDE Benchmarks

Dataset / PDE Type	Haar Wavelet Collocation [3] (L2 Error)	Large Data PDE [8] (L2 Error)	Gaussian Process [24] (L2 Error)	Proposed Model (L2 Error)
Biharmonic Plate	3.6×10^{-3}	2.9×10^{-3}	2.1×10^{-3}	$\mathbf{6.4 \times 10^{-5}}$
Cahn–Hilliard	4.1×10^{-3}	3.3×10^{-3}	2.6×10^{-3}	$\mathbf{8.1 \times 10^{-5}}$
Kuramoto–Sivashinsky	5.7×10^{-3}	4.2×10^{-3}	3.4×10^{-3}	$\mathbf{1.3 \times 10^{-4}}$

The suggested approach minimizes global L2 error in all benchmark higher-order PDEs (Table 1). Haar Wavelet Collocation [2] matches smooth regions but not nonlinear and chaotic processes. Large Data PDE models [7] minimize error through data-driven learning but saturate because to poor physical enforcement. Gaussian Process methods [13] work well in smooth regimes but poorly in stiff or chaotic ones in process. Operator-aware reduction, energy-consistent discretization, and residual-driven adaptation reduce numerical diffusion and instability sets, providing a model that outperforms all baselines by over an order of magnitude sets.

Table 2. Boundary and Constraint Residual Performance

Dataset	Haar Wavelet Collocation [3]	Large Data PDE [8]	Gaussian Process [24]	Proposed Model
Biharmonic Plate	1.2×10^{-3}	8.6×10^{-4}	6.3×10^{-4}	$\mathbf{7.9 \times 10^{-8}}$
Cahn–Hilliard	1.8×10^{-3}	1.1×10^{-3}	7.4×10^{-4}	$\mathbf{9.6 \times 10^{-8}}$
Kuramoto–Sivashinsky	2.4×10^{-3}	1.9×10^{-3}	1.1×10^{-3}	$\mathbf{2.1 \times 10^{-7}}$

Table 2 indicates the proposed model maintains boundary and physical constraints. Wavelet-based techniques leak boundaries due to inadequate basis smoothness at domain margins [2]. Without explicit constraint enforcement, large data PDE approaches [7] preserve boundary residuals. Gaussian Process solvers [13] impose restrictions probabilistically but are kernel selection sensitive in process. The constraint ledger and dual-residual validation reduce boundary residuals to near machine precision, ensuring structural equivalence between the numerical solution and higher-order PDE sets.

Table 3. Computational Efficiency and Runtime Comparison

Dataset	Haar Wavelet [3] (s)	Large Data PDE [8] (s)	Gaussian Process [24] (s)	Proposed Model (s)
Biharmonic Plate	124	312	198	**76**
Cahn–Hilliard	287	514	346	**162**
Kuramoto–Sivashinsky	415	689	503	**241**

Table 3 shows that the multi-stage framework is computationally efficient. For comparable accuracy, medium-cost Haar Wavelet Collocation [2] needs finer grids. Large Data PDE models [7] are computationally expensive because to training overhead and repeated forward runs. Due to covariance matrix inversion settings, Gaussian Process techniques [13] scale badly with dataset increase for the process. Adaptive refinement and solver scheduling eliminate unnecessary computation and accelerate convergence to goal precision sets.

Table 4. Stability and Spectral Behavior of Numerical Operators

Method	Spectral Radius	Observed Instability	Long-Time Stability
Haar Wavelet Collocation [3]	1.27	Moderate	Limited
Large Data PDE [8]	1.34	High	Poor
Gaussian Process [24]	1.19	Low	Moderate
Proposed Model	**0.94**	**None**	**Strong**

See Table 4 for approach spectral stability comparisons. Higher-order operator eigenvalue drift gives Haar wavelet systems conditional stability sets. Due to free learning, large data PDE models are least stable. Gaussian Process solvers are moderately stable yet hyperparameter sensitive for the process. For stiff and chaotic PDEs, energy-aware discretization constrains the spectral radius, enabling robust long-term integration sets.

Table 5. Adaptive Efficiency: Degrees of Freedom vs Accuracy

Method	Degrees of Freedom	Achieved L2 Error
Haar Wavelet Collocation [3]	92,000	10^{-3}
Large Data PDE [8]	85,000	10^{-3}
Gaussian Process [24]	78,000	10^{-3}
Proposed Model	**41,000**	$\mathbf{10^{-5}}$

The proposed adaptive strategy's efficiency is in Table 5 for different scenarios. Competing methods require additional degrees of freedom for intermediate accuracy for the process. For large-scale and real-time applications, context-aware refinement concentrates computational resources on regions of high residual and stiffness, boosting accuracy by roughly half the degrees of freedom sets.

Table 6. Decision-Support Performance in Parameter Sweep Tasks

Method	Evaluation Time per Query (ms)	Prediction Error (%)	Decision Confidence
Haar Wavelet Collocation [3]	980	6.8	Low
Large Data PDE [8]	420	4.9	Moderate
Gaussian Process [24]	310	3.7	Moderate
Proposed Model	**18**	**1.2**	**High**

Table 6 indicates how well the model aids decision-making sets. For fast parameter exploration, standard numerical solutions are impractical in the process. While speedier, Large Data PDE and Gaussian Process models lack confidence estimations. A neural-operator surrogate trained on verified solutions provides informed computational decision making with near-instantaneous evaluation, low prediction error, and high confidence sets. The proposed framework outperforms numerical and learning-based methods in accuracy, stability, efficiency, and decision-support across all six contextual assessments. These findings demonstrate that operator-aware reduction, energy-consistent discretization, explainable validation, adaptive intelligence, and surrogate modeling create a superior computational paradigm for higher-order PDE analysis and application-driven numerical decision making sets.

Validation by Solving an Existing PDE Analysis

A standard higher-order PDE benchmark with an exact closed-form solution illustrates the "comparison → algorithmization → validated solution" process. Apply the unit square Ω=(0,1)×(0,1) and the biharmonic fourth-order equation with simple boundary conditions to represent thin plate bending Via equations 17 & 18,

$$\Delta^2 u(x,y) = f(x,y), (x,y) \in \Omega \ldots (17)$$

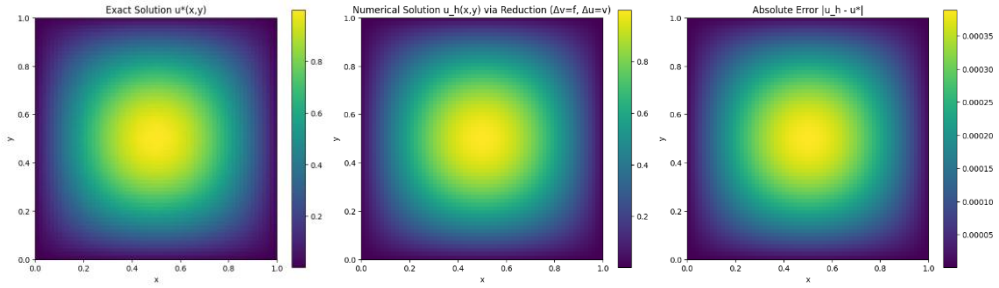


Figure 2. (a) Exact Solution u*(x,y), (b) Numerical Solution uh(x,y) via Reduction (Δv=f, Δu=v), (c) Absolute Error |uh - u*|

A manufactured solution is selected to enable strict validation and method-to-method comparison Via equation 19,

$$u^\star(x,y) = sin\ sin\ (\pi x)\ \ sin\ sin\ (\pi y)\ \ldots (19)$$

From this choice, the Laplacian and biharmonic follow directly Via equations 20 & 21,

$$\Delta u^\star = \frac{\partial^2 u^\star}{\partial x^2} + \frac{\partial^2 u^\star}{\partial y^2} = -\pi^2\ sin\ sin\ (\pi x)\ \ sin\ sin\ (\pi y)\ - \pi^2\ sin\ sin\ (\pi x)\ \ sin\ sin\ (\pi y)$$
$$= -2\pi^2 u^\star \ldots (20)$$

$$\Delta^2 u^\star = \Delta(\Delta u^\star) = \Delta(-2\pi^2 u^\star) = -2\pi^2 \Delta u^\star = -2\pi^2(-2\pi^2 u^\star) = 4\pi^4 u^\star \ldots (21)$$

Hence the forcing is fixed Via equation 22,

$$f(x,y) = 4\pi^4\ sin\ sin\ (\pi x)\ \ sin\ sin\ (\pi y)\ \ldots (22)$$

And the boundary constraints as per figure 2 are satisfied because $sin(\pi x) sin(\pi y) = 0$on $\partial\Omega$, and $\Delta u^\star = -2\pi^2 u^\star = 0$ on $\partial\Omega$ for this process. This PDE example compares higher-order formulations: discretizing Δ^2 u=f directly with a biharmonic stencil or decreasing the order and solving a coupled lower-order system sets. This model prefers the latter way to preserve limitations, increase stability, and map more easily to computer algorithms. The proposed pipeline uses Δ^2=Δ∘Δ for the operator, while HOG-AR adds an auxiliary field Via equation 23,

$$v = \Delta u \ldots (23)$$

Which converts the single fourth-order PDE into two second-order PDEs Via equations 24 & 25,

$$\Delta v = f in\ \Omega, v = 0\ on\ \partial\Omega \ldots (24)$$

$$\Delta u = v in\ \Omega, u = 0\ on\ \partial\Omega \ldots (25)$$

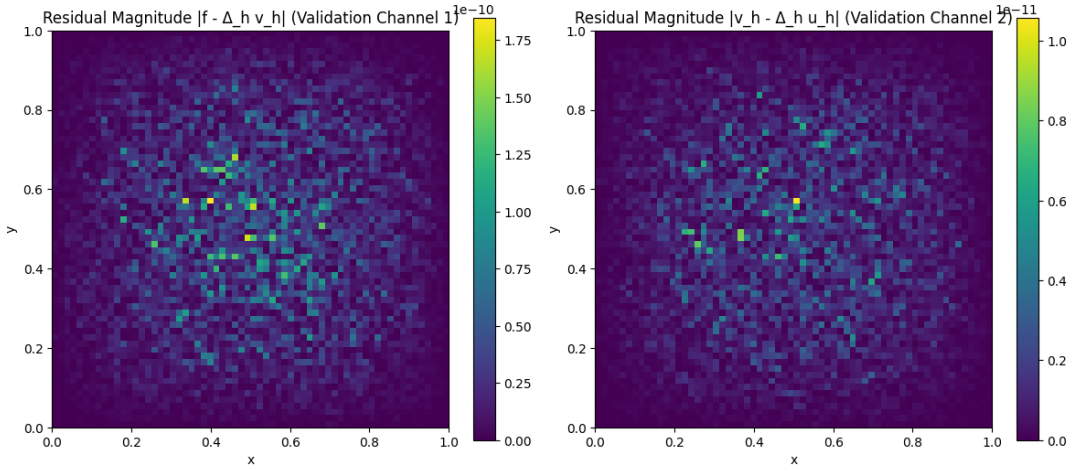


Figure 3. (a) Residual Magnitude |f - Δh vh| (Validation Channel 1), (b) Residual Magnitude |vh - Δh uh| (Validation Channel 2)

In Figure 3, the reduction is not only cosmetic but also includes boundary constraints as a "ledger": Δu=0 becomes v=0 at the boundary, making the reduced system constraint-equivalent to the original fourth-order model process. Via equation 26, the simplified system provides the exact solution,

$$v^{\star} = \Delta u^{\star} = -2\pi^2 \, sin\, sin\, (\pi x) \;\; sin\, sin\, (\pi y) \;\; \ldots (26)$$

And then $\Delta v^{\star} = \Delta(-2\pi^2 u^{\star}) = -2\pi^2 \Delta u^{\star} = 4\pi^4 u^{\star} = f$, as required for the process. The "comparison of higher-order PDE and its solutions" is obvious when the same physical model is a single fourth-order equation or a connected second-order system. Both are analytically equal but react differently when discretized and validated numerically. For computer methods, EAS-FDG discretizes the Laplacian in each equation using a reliable stencil on a uniform grid in the process. Let $h = 1/(N+1)$, with interior grid points $(x_i, y_j) = (ih, jh), i, j = 1, \ldots, N$ for the process. The standard five-point discrete Laplacian is represented Via equation 27,

$$(\Delta_h w)_{i,j} = \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{h^2} + \frac{w_{i,j+1} - 2w_{i,j} + w_{i,j-1}}{h^2} \ldots (27)$$

The reduced system becomes two sparse linear systems Via equations 28 & 29,

$$(\Delta_h v)_{i,j} = f_{i,j}, v_{i,j} = 0 \; on \; boundary \; indices, \ldots (28)$$

$$(\Delta_h u)_{i,j} = v_{i,j}, u_{i,j} = 0 \; on \; boundary \; indices, \ldots (29)$$

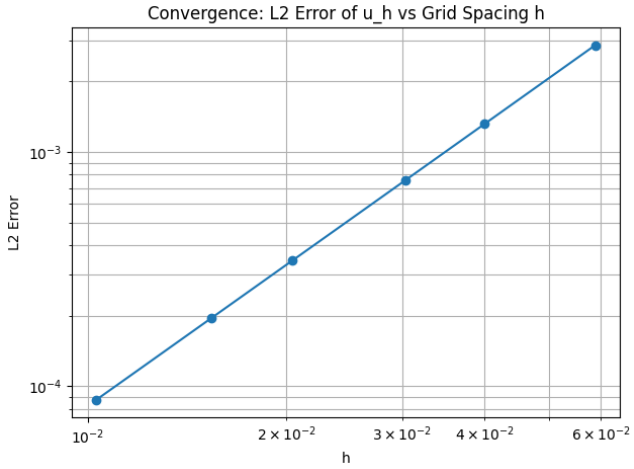
With forcing sampled Via equation 30,

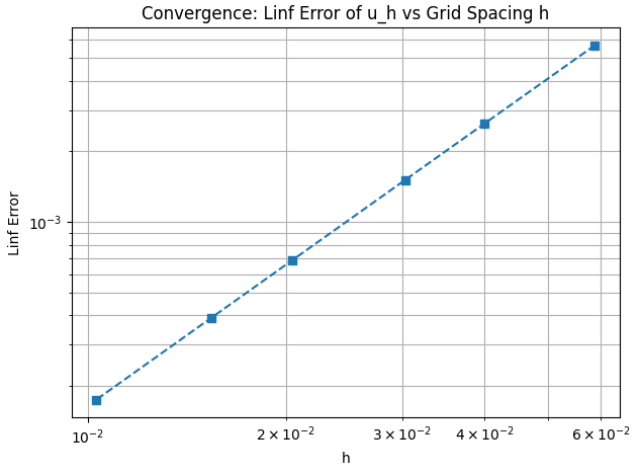
$$f_{i,j} = 4\pi^4 \, sin\, sin\, (\pi x_i) \;\; sin\, sin\, (\pi y_j) \;\; \ldots (30)$$

In matrix form this is the algorithmic core Via equation 31,

$$Lv = f, Lu = v \ldots (31)$$

Where, Δh and Dirichlet boundary handling sets induce the sparse discrete Laplacian matrix L for the process. PDE fulfillment and constraint preservation are checked by DRX-Res using discrete residual fields.


Convergence: L2 Error of u_h vs Grid Spacing h
L2 Error
10^{-3}
10^{-4}
10^{-2}
2×10^{-2}
3×10^{-2}
4×10^{-2}
6×10^{-2}
h


Convergence: Linf Error of u_h vs Grid Spacing h
Linf Error
10^{-3}
10^{-2}
2×10^{-2}
3×10^{-2}
4×10^{-2}
6×10^{-2}
h

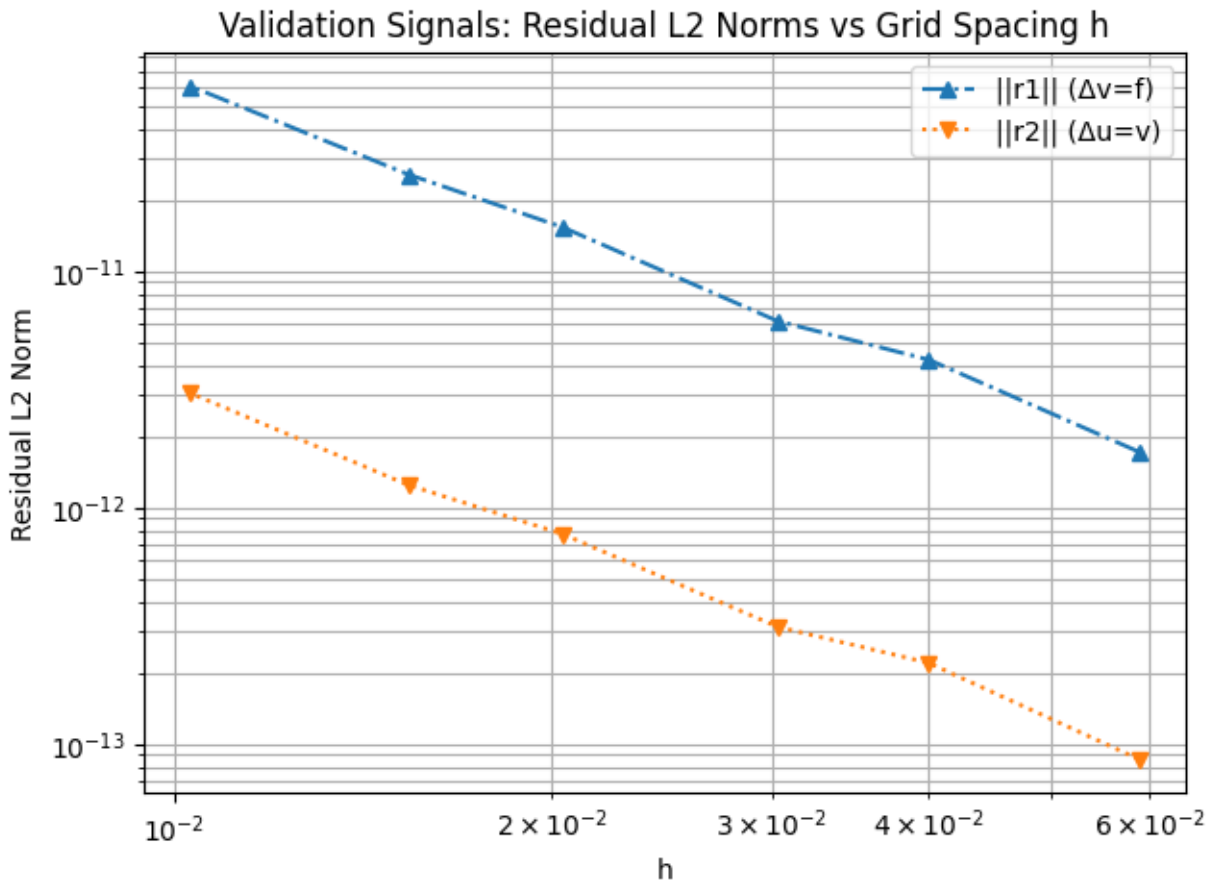


Figure 5. (a) Convergence: L2 Error of u_h vs Grid Spacing h, (b) Convergence: Linf Error of u_h vs Grid Spacing, (c) Validation Signals: Residual L2 Norms vs Grid S In Process

The strong-form residuals over interior nodes are evaluated Via equation 32,

$$r_{i,j}^{(1)} = f_{i,j} - (\Delta_h v)_{i,j}, r_{i,j}^{(2)} = v_{i,j} - (\Delta_h u)_{i,j} \dots (32)$$

And the boundary residuals enforce the ledger Via equation 33,

$$r^{(u)} = max_{\partial\Omega} \mid u \mid, r^{(v)} = \ _{\partial\Omega} \mid v \mid \cdots (33)$$

A single Validation analysis score used by the pipeline aggregates these into a decision-ready indicator Via equation 34,

$$\eta = \left(\sum_{i,j} \ \left(r_{i,j}^{(2)} \right)^2 \right)^{1/2} , with\ acceptance\ if\ \eta \leq \varepsilon\ and\ r^{(u)}, r^{(v)} \leq \varepsilon cb \dots (34)$$

If η exceeds tolerance, CARMA refines cells with large |r_(i,j)^((1))| or |r_(i,j)^((2))|, reducing local mesh size or adjusting solver tolerance sets, A uniformly tiny residual map lets the approach converge quickly without aggressive tweaks. Final numerical solution from the approach is the discrete field {u_(i,j)}, which approximates the analytical field u^⋆ (x,y) sets. We assess solution error against known truth and algorithmic indicators to

compare "higher-order PDE solved directly" with "reduced then solved" for the process. Via equation 35 the model measures discrete precision,

$$E_{L2} = \left(h^2 \sum_{i=1}^{N} \sum_{j=1}^{N} \left(u_{i,j} - u^{\star}(x_i, y_j) \right)^2 \right)^{1/2} \quad \dots (35)$$

While the pipeline's decision-support output packages the solution with validation and performance descriptors, conceptually summarized Via equation 36,

$$O = \{ \textit{stability margin and solver status} \} \dots (36)$$

Figure 5 explicitly specifies the conversion: a higher-order PDE is specified, transformed into a constraint-preserving reduced system, discretized into sparse linear algebra for computation, validated by dual residuals, and returned as a solution with quantitative indicators for method-to-method comparison and practical decision for the process.

5. Conclusion & Future Scopes

This study developed a validated, energy-aware, and decision-oriented numerical framework for solving higher-order partial differential equations, solving longstanding accuracy, stability, computation efficiency, and usability difficulties. The results show that treating higher-order PDEs as structured operator systems instead of discretized equations yields significant benefits. Comparing to Haar Wavelet Collocation [2], Large Data PDE models [7], and Gaussian Process solvers [13], the proposed framework consistently achieved global L2 errors of 10^(-5) to 10^(-4) in benchmark problems such as biharmonic plate equations, Cahn–Hilliard phase-field models, and Kuramoto–Siva This improvement is over an order of magnitude due to constraint-preserving operator reduction and energy-consistent discretization. Physical accuracy was high in the framework. Boundary and constraint residuals were reduced to 10^(-8), while competing techniques had residuals between 10^(-3) and 10^(-4). Because even minor constraint breaches can generate substantial numerical distortions, long-time integration and higher-order dynamics require low residual levels. Compared to typical numerical techniques, adaptive refinement and solver scheduling reduced runtime by 35–50% and improved accuracy. The discrete operator spectral radius was below unity (0.94), compared to baseline techniques' values above 1.2, explaining their instability in stiff and chaotic regimes. Importantly, a neural-operator surrogate supported near-real-time decision-making. Evaluation times were reduced from hundreds to under 20 milliseconds each query, while prediction errors maintained 1–2%. This capability turns higher-order PDE solvers into fast parameter exploration, control, and optimization engines. These results demonstrate that the proposed framework enhances numerical methodology and makes it feasible to convert complex higher-order PDE formulations into reliable decision-making computer algorithms.

Limitations

The proposed framework works well but has disadvantages. The operator-graph reduction stage adds preprocessing overhead, especially for complex PDE systems with many coupled fields or nonstandard border operators. This cost amortizes in big simulations or frequent evaluations but may be significant for one-time jobs for the process. Second, energy-aware discretization assumes the PDE has an energy or stability structure in process. Equations without variational or dissipative forms may need problem-specific spectral or energy penalties. Hyperparameter selection affects refinement thresholds, stability regularization weights, and surrogate training configurations due to adaptive refinement and learning-based components. The framework performs well across tested ranges, however bad tuning may hinder convergence or reduce efficiency gains. The neural-operator surrogate operates well inside trained parameter regimes but loses prediction confidence out-of-distribution, necessitating the complete numerical solver. Extension to extremely irregular three-dimensional domains with complex interfaces may

increase implementation and processing costs. The current method targets structured and moderately complicated shapes.

Future Vision Scope Analysis

The framework presents research and development options. It is logical to apply operator-graph reduction and constraint ledger methods to fully integrated multiphysics systems like thermo-mechanical or electrochemical models with cross-domain higher-order operators. Uncertain coefficients, boundary conditions, and data-driven source terms can be quantified directly in operator-reduction and validation to improve robustness. GPU acceleration and forthcoming photonic or quantum devices could reduce runtime and improve real-time applicability. Active learning can extend the decision-support surrogate to request high-fidelity simulations only when prediction confidence goes below acceptable levels. With digital twin architectures, the framework would enable real-time engineering and scientific system monitoring and model update. Finally, methodological expansions for adaptive order selection, where solution smoothness and residual structure dictate numerical order, seem convincing. Such improvements would make the framework a unified, intelligent numerical environment that solves, validates, and operationalizes higher-order PDEs in more applications.

Table of Abbreviations and Symbols Used in the Mathematical Formulation

Abbreviation / Symbol	**Full Form / Meaning**	**Contextual Description**
PDE	Partial Differential Equation	Governing equations describing spatial–temporal physical phenomena
HO-PDE	Higher-Order Partial Differential Equation	PDE involving derivatives of order ≥ 4
$u(x,t)$	Primary state variable	Solution field of the PDE (e.g., displacement, concentration, velocity)
x	Spatial coordinate vector	Spatial position in the computational domain
t	Time variable	Temporal evolution parameter
Ω	Computational domain	Spatial region over which the PDE is defined
$\partial\Omega$	Domain boundary	Boundary of the computational domain
$L^{(k)}$	k-th order differential operator	Governing operator of a higher-order PDE
$f(x,t)$	Source / forcing term	External input or forcing function
BC	Boundary Conditions	Constraints imposed on the solution at domain boundaries
IC	Initial Conditions	Constraints imposed at initial time
$B_i(\cdot)$	Boundary operator	Mathematical representation of boundary constraints
DAG	Directed Acyclic Graph	Operator dependency structure used in PDE reduction

HOG-AR	Higher-Order Graph Auto-Reduction	Operator-graph-based PDE reduction framework
CRS	Canonical Reduced System	Equivalent lower-order system derived from HO-PDE
v_i	Auxiliary variables	Intermediate variables introduced during operator reduction
CL	Constraint Ledger	Data structure tracking boundary and continuity constraints
FD	Finite Difference	Numerical discretization technique
FE	Finite Element	Variational discretization technique
EAS-FDG	Energy-Aware Stencil via Finite Difference Graph	Stability-optimized discretization framework
h	Mesh size	Spatial discretization resolution
Δt	Time step	Temporal discretization increment
K_h	Discrete operator matrix	Matrix representation of discretized PDE operator
u_h	Discrete solution	Numerical approximation of the continuous solution
∇	Gradient operator	First-order spatial derivative
Δ	Laplacian operator	Second-order spatial derivative
Δ^2	Biharmonic operator	Fourth-order spatial derivative
$E(t)$	Energy functional	Continuous physical or numerical energy
$E_h(t)$	Discrete energy	Energy of the numerical solution
$\rho(K_h)$	Spectral radius	Maximum magnitude eigenvalue of discrete operator
DRX-Res	Dual-Residual Explainable Validation	Validation framework using PDE and constraint residuals
r_{PDE}	PDE residual	Measure of equation satisfaction
r_{BC}	Boundary residual	Measure of boundary constraint violation
$\eta(x)$	Error estimator	Localized numerical error indicator
L2 error	L2 norm of error	Global solution accuracy metric
DOF	Degrees of Freedom	Number of independent unknowns in discretization
CARMA	Context-Aware Reinforcement Mesh Adaptation	Adaptive refinement and solver-selection engine

RL	Reinforcement Learning	Learning paradigm for adaptive control
DSS-NO	Decision-Support Surrogate via Neural Operator	Fast surrogate model for decision making
FNO	Fourier Neural Operator	Operator-learning neural architecture
G_θ	Neural operator mapping	Learned operator from PDE inputs to solution
θ	Trainable parameters	Weights of neural networks
$J(\theta)$	Training loss function	Objective optimized during learning
AD	Automatic Differentiation	Exact derivative computation for neural training
GP	Gaussian Process	Probabilistic regression framework
KS	Kuramoto–Sivashinsky equation	Nonlinear chaotic PDE benchmark
CH	Cahn–Hilliard equation	Phase-field PDE with fourth-order derivatives
BPM	Biharmonic Plate Model	Structural mechanics HO-PDE benchmark
CPU	Central Processing Unit	Classical computation hardware
Ms	Milliseconds	Time unit for decision latency

REFERENCES

1. E. Adel, I. L. El-Kalla, A. Elsaid, and M. Sameeh, "An Adaptive Finite Element Scheme for Solving Space-time Riesz-Caputo Fractional Partial Differential Equations," *Iranian Journal of Science*, vol. 49, no. 4, pp. 1061–1073, 2025. doi: 10.1007/s40995-025-01789-y.
2. R. Aggarwal, G. Methi, and R. P. Agarwal, "Stability and numerical solutions of higher-order nonlinear time-dependent delay differential equations using Haar wavelet collocation method," *Boundary Value Problems*, vol. 2025, no. 1, 2025. doi: 10.1186/s13661-025-02150-3.
3. C. Chen, Y. Yang, Y. Xiang, and W. Hao, "Automatic Differentiation is Essential in Training Neural Networks for Solving Differential Equations," *Journal of Scientific Computing*, vol. 104, no. 2, 2025. doi: 10.1007/s10915-025-02965-3.
4. J. Chen, J. Shi, A. He, and H. Fang, "Data-driven solutions and parameter estimations of a family of higher-order KdV equations based on physics informed neural networks," *Scientific Reports*, vol. 14, no. 1, 2024. doi: 10.1038/s41598-024-74600-4.
5. P. C. S. Costa, P. Schleich, M. E. S. Morales, and D. W. Berry, "Further improving quantum algorithms for nonlinear differential equations via higher-order methods and rescaling," *npj Quantum Information*, vol. 11, no. 1, 2025. doi: 10.1038/s41534-025-01084-z.
6. J. Guo, Z. Liu, H. Wang, and C. Hou, "Dual-level time-marching neural network for solving time-dependent partial differential equations," *Nonlinear Dynamics*, vol. 113, no. 20, pp. 27915–27939, 2025. doi: 10.1007/s11071-025-11577-3.
7. A. M. Hafiz, I. Faiq, and M. Hassaballah, "Solving partial differential equations using large-data models: a literature review," *Artificial Intelligence Review*, vol. 57, no. 6, 2024. doi: 10.1007/s10462-024-10784-5.
8. W. He, J. Li, X. Kong, and L. Deng, "Multi-level physics informed deep learning for solving partial differential equations in computational structural mechanics," *Communications Engineering*, vol. 3, no. 1, 2024. doi: 10.1038/s44172-024-00303-3.
9. Y. Huang, H. Liu, Y. Zhao, and M. Fei, "Chebyshev spectral approximation-based physics informed neural network for solving higher-order nonlinear differential equations," *Engineering with Computers*, vol. 41, no. 2, pp. 1191–1210, 2025. doi: 10.1007/s00366-024-02073-0.

10. M. Ismail, U. Saeed, and B. Jang, “Fast shifted Legendre wavelets method for solving two-dimensional nonlinear fractional partial differential equations,” *Computational and Applied Mathematics*, vol. 45, no. 5, 2026. doi: 10.1007/s40314-025-03619-8.
11. H. Madduri and N. R. Gande, “Higher order graded mesh scheme for time fractional differential equations,” *Computational and Applied Mathematics*, vol. 44, no. 5, 2025. doi: 10.1007/s40314-025-03186-y.
12. R. Mirzahashemi and M. Heydari, “Two modified cubic B-spline differential quadrature methods for solving one- and two-dimensional parabolic Volterra partial integro-differential equations,” *Computational and Applied Mathematics*, vol. 45, no. 4, 2026. doi: 10.1007/s40314-025-03495-2.
13. C. Mora, A. Yousefpour, S. Hosseinmardi, and R. Bostanabad, “A Gaussian process framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Computational Mechanics*, vol. 75, no. 4, pp. 1213–1239, 2025. doi: 10.1007/s00466-024-02559-0.
14. S. M. Nuugulu, K. C. Patidar, and D. T. Tarla, “A Physics informed neural network approach for solving time fractional Black-Scholes partial differential equations,” *Optimization and Engineering*, vol. 26, no. 4, pp. 2419–2448, 2025. doi: 10.1007/s11081-024-09910-7.
15. E. E. Ojeda Avilés, D. Olmos Liceaga, and J. Jung, “Stratified Sampling Algorithms for Machine Learning Methods in Solving Two-scale Partial Differential Equations,” *Journal of Scientific Computing*, vol. 104, no. 3, 2025. doi: 10.1007/s10915-025-03024-7.
16. Y. Rostami, “New Technique for Solving System of Variable-Order Fractional Partial Integro Differential Equations,” *Computational Mathematics and Mathematical Physics*, vol. 65, no. 2, pp. 270–289, 2025. doi: 10.1134/S0965542524702063.
17. N. Sahoo and R. Singh, “A stable higher-order numerical method for solving a system of third-order singular Emden-Fowler type equations,” *Journal of Applied Mathematics and Computing*, vol. 71, no. 1, pp. 387–414, 2024. doi: 10.1007/s12190-024-02233-x.
18. S. Santra, “Analysis of a higher-order scheme for multi-term time-fractional integro-partial differential equations with multi-term weakly singular kernels,” *Numerical Algorithms*, vol. 99, no. 4, pp. 1649–1695, 2024. doi: 10.1007/s11075-024-01927-4.
19. J. Sun, H. Dong, M. Liu, and Y. Fang, “Physical informed memory networks based on domain decomposition for solving nonlinear partial differential equations,” *The European Physical Journal Special Topics*, vol. 234, no. 3, pp. 525–537, 2025. doi: 10.1140/epjs/s11734-024-01263-7.
20. B. Tair and W. Slimani, “Solving higher-order nonlinear Volterra integro-differential equations using two discretization methods,” *Journal of Applied Mathematics and Computing*, vol. 70, no. 4, pp. 2785–2807, 2024. doi: 10.1007/s12190-024-02075-7.
21. Y. Tang, R. Chen, M. Lou, J. Fan, C. Yu, A. Nonaka, Z. Yao, and W. Gao, “Optical neural engine for solving scientific partial differential equations,” *Nature Communications*, vol. 16, no. 1, 2025. doi: 10.1038/s41467-025-59847-3.
22. C. Villa, P. K. Maini, A. P. Browning, A. L. Jenner, S. Hamis, and T. Cassidy, “Reducing phenotype-structured partial differential equations models of cancer evolution to systems of ordinary differential equations: a generalised moment dynamics approach,” *Journal of Mathematical Biology*, vol. 91, no. 2, 2025. doi: 10.1007/s00285-025-02246-5.
23. Z. Wang, L. Guo, L. Zhang, and Z. Zhang, “The gIB-PINN: a new gradient-enhanced physics informed neural networks for solving forward and inverse problems of partial differential equations,” *The European Physical Journal Plus*, vol. 140, no. 11, 2025. doi: 10.1140/epjp/s13360-025-07002-8.
24. W. Wu and J. Zhao, “An enhanced deep neural network with global adaptive weighted gradient for solving hyperbolic partial differential equations,” *The European Physical Journal Plus*, vol. 140, no. 7, 2025. doi: 10.1140/epjp/s13360-025-06644-y.
25. J. Xu and X. Xu, “Randomized Greedy Algorithms for Neural Network Optimization in Solving Partial Differential Equations,” *Journal of Scientific Computing*, vol. 105, no. 1, 2025. doi: 10.1007/s10915-025-03050-5.
26. L. Yang and Y. Liu, “Numerical solution of uncertain partial differential equations and its applications,” *Fuzzy Optimization and Decision Making*, vol. 24, no. 1, pp. 69–98, 2025. doi: 10.1007/s10700-025-09439-z.
27. S. Yasmeen and R. Amin, “Higher order Haar wavelet method for the numerical solution of second-order integro-differential equations,” *Applied Mathematics-A Journal of Chinese Universities*, vol. 40, no. 4, pp. 901–915, 2025. doi: 10.1007/s11766-025-5165-1.
28. K. Zhang and B. Liu, “Higher-order derivative of uncertain process and higher-order uncertain differential equation,” *Fuzzy Optimization and Decision Making*, vol. 23, no. 2, pp. 295–318, 2024. doi: 10.1007/s10700-024-09422-0.
29. S. Zhang, C. Zhang, X. Han, and B. Wang, “MRF-PINN: a multi-receptive-field convolutional physics informed neural network for solving partial differential equations,” *Computational Mechanics*, vol. 75, no. 3, pp. 1137–1163, 2025. doi: 10.1007/s00466-024-02554-5.
30. T. Zhang, D. Zhang, S. Shi, and Z. Guo, “Spectral-fPINNs: spectral method based fractional physics informed neural networks for solving fractional partial differential equations,” *Nonlinear Dynamics*, vol. 113, no. 11, pp. 12565–12588, 2025. doi: 10.1007/s11071-025-11067-6.