MANTECH Publications

Journal of Engineering Mathematics & Statistics
Volume 5, Issue 1, January-June 2021

Machine Learning Enhanced Numerical Solvers

Sailendra Tyagi[1], Chukesh Mishra[2]

Associate Professor[1], Students[2]

Department of Applied Mathematics

Horizon College of Science, India

***Email ID**: Sailendratyagi55@gmail.com[1], chukesh7m0mishra@yahoo.com[2]*

Abstract

The increasing complexity of scientific and engineering problems necessitates the development of efficient and accurate numerical solvers. Traditional numerical solvers, while reliable, often face challenges with high-dimensional systems, nonlinearity, and computational cost. Recently, the integration of machine learning (ML) techniques into numerical methods has emerged as a promising approach to enhance solver performance. This review presents a comprehensive analysis of machine learning-enhanced numerical solvers, focusing on their principles, methodologies, applications, and limitations. Techniques such as neural network-based solvers, physics-informed neural networks, and surrogate models are discussed in detail. Additionally, the paper explores comparative studies demonstrating the computational advantages and accuracy improvements achieved by ML-assisted solvers. The review concludes by highlighting current challenges, future research directions, and potential interdisciplinary applications.

Keywords: *Machine learning, numerical solvers, physics-informed neural networks, surrogate models, computational efficiency, high-dimensional systems*

INTRODUCTION

Numerical solvers are the backbone of computational simulations in engineering, physics, and applied mathematics. Classical methods, such as finite difference methods (FDM), finite element methods (FEM), and spectral methods, have been successfully applied to solve ordinary differential equations (ODEs), partial differential equations (PDEs), and large systems

Page 39-51 © MANTECH PUBLICATIONS 2021. All Rights Reserved

of algebraic equations. However, these methods often encounter limitations when applied to large-scale or highly nonlinear systems, as they demand substantial computational resources and may suffer from stability or convergence issues.

Machine learning (ML) has emerged as a transformative tool in computational science. By learning from data, ML algorithms can approximate complex mappings, reduce computational cost, and enhance the accuracy of simulations. Integrating ML into traditional numerical solvers provides a hybrid approach, leveraging the robustness of classical solvers and the adaptability of data-driven methods. This review focuses on the current state-of-the-art in machine learning-enhanced numerical solvers, emphasizing methodologies, applications, advantages, and challenges.

BACKGROUND ON NUMERICAL SOLVERS

Numerical solvers are fundamental tools in computational science, enabling the approximation of solutions to mathematical problems that are analytically intractable. These problems frequently arise in engineering, physics, finance, and applied mathematics. Numerical solvers can be broadly classified into methods for solving **linear and nonlinear algebraic systems** and methods for solving **ordinary and partial differential equations**. Understanding the principles, advantages, and limitations of these classical methods is essential before exploring how machine learning can enhance their performance.

Classical Numerical Methods

Finite Difference Method (FDM)

The Finite Difference Method (FDM) is one of the simplest and most widely used approaches for approximating derivatives. In FDM, derivatives in differential equations are replaced by differences between function values at discrete grid points. For example, the first derivative of a function u(x)u(x)u(x) at point xix_ixi can be approximated using:

dudx|xi≈ui+1−uiΔx(forward difference)\frac{du}{dx} \bigg|_{x_i} \approx \frac{u_{i+1} - u_i}{\Delta x} \quad (\text{forward difference})dxduxi≈Δxui+1−ui(forward difference)

Page 39-51 © MANTECH PUBLICATIONS 2021. All Rights Reserved

Similarly, central and backward differences offer higher accuracy. FDM is particularly effective for problems defined on **regular, structured grids** and is often applied to heat conduction, diffusion, and wave propagation problems.

Advantages:

- Simple to implement.
- Suitable for time-dependent and steady-state problems.
- Compatible with explicit and implicit time integration schemes.

Limitations:

- Stability constraints such as the Courant–Friedrichs–Lewy (CFL) condition can limit time-step sizes.
- Difficulties arise in complex geometries due to its reliance on structured grids.

Finite Element Method (FEM)

The Finite Element Method (FEM) addresses some limitations of FDM by discretizing the problem domain into smaller **elements**, over which approximate solutions are constructed using basis (shape) functions. FEM is highly versatile and can handle irregular geometries and complex boundary conditions. The general approach involves formulating the **weak form** of the governing equation, assembling the global system of equations, and solving for nodal values.

Applications:

- Structural mechanics: stress-strain analysis of beams, plates, and shells.
- Fluid dynamics: incompressible and compressible flows.
- Heat transfer: conduction, convection, and phase change problems.

Advantages:

- Handles complex geometries and boundary conditions.
- Provides high accuracy for problems with localized features.
- Well-established software and libraries are available (e.g., ANSYS, COMSOL, Abaqus).

Limitations:

- Assembling and solving large systems can be computationally expensive.
- Requires mesh refinement for accuracy, increasing memory usage and computational cost.

Page 39-51 © MANTECH PUBLICATIONS 2021. All Rights Reserved

Spectral Methods

Spectral methods solve differential equations by expanding the solution as a sum of orthogonal basis functions, such as Fourier series or Chebyshev polynomials:

u(x)≈∑k=0Nakϕk(x)u(x) \approx \sum_{k=0}^{N} a_k \phi_k(x)u(x)≈k=0∑Nakϕk(x)

where aka_kak are the coefficients and ϕk(x)\phi_k(x)ϕk(x) are the basis functions. Spectral methods are **highly accurate for smooth solutions**, often achieving exponential convergence.

Applications:

- Fluid flow simulations with smooth velocity fields.
- Wave propagation problems.
- Quantum mechanics and periodic domains.

Advantages:

- Extremely high accuracy for smooth problems.
- Efficient for global solutions with few degrees of freedom.

Limitations:

- Struggle with non-smooth solutions and discontinuities (Gibbs phenomenon).
- Complex implementation for irregular domains or complex boundary conditions.

Iterative Solvers

Iterative solvers are essential for **large, sparse linear systems** that arise in FEM, FDM, and other discretizations. Common methods include:

- **Jacobi method**: Updates each variable based on previous iteration values.
- **Gauss-Seidel method**: Uses the latest updated values immediately within the same iteration.
- **Conjugate Gradient (CG) method**: Efficient for symmetric positive-definite systems.
- **GMRES (Generalized Minimal Residual)**: Suitable for non-symmetric systems.

Advantages:

- Memory-efficient for sparse systems.
- Can handle very large systems that direct solvers cannot.

Limitations:

Page 39-51 © MANTECH PUBLICATIONS 2021. All Rights Reserved

- Convergence can be slow for ill-conditioned matrices.
- Preconditioning is often required to improve efficiency.

2.2 Limitations of Classical Solvers

While classical numerical solvers are highly reliable and mathematically rigorous, they have several inherent limitations, particularly when applied to modern large-scale and complex problems:

1. **High Computational Cost:**

   Fine spatial and temporal discretization significantly increases the number of unknowns, resulting in high memory usage and long computation times. For example, a 3D FEM simulation with millions of elements can take hours or even days on standard hardware.

2. **Nonlinearity:**

   Many practical systems are governed by nonlinear PDEs (e.g., Navier–Stokes equations). Solving these requires iterative linearization methods such as Newton-Raphson, which can be computationally expensive and sensitive to initial guesses.

3. **Dimensionality:**

   High-dimensional systems, such as multi-physics simulations or stochastic PDEs, suffer from the **curse of dimensionality**, where the number of computational points grows exponentially with the number of dimensions. This severely limits efficiency and feasibility.

4. **Parameter Sensitivity and Uncertainty:**

   Classical solvers are sensitive to input parameters. To account for uncertainties, multiple simulations (Monte Carlo methods) are often required, further increasing computational demands.

5. **Complexity in Multiscale Problems:**

   Problems spanning multiple spatial or temporal scales require adaptive meshes or multiscale methods. Classical solvers may struggle to maintain accuracy and stability across scales without significant computational overhead.

Page 39-51 © MANTECH PUBLICATIONS 2021. All Rights Reserved

MACHINE LEARNING INTEGRATION IN NUMERICAL SOLVERS

The increasing complexity of computational problems has motivated the integration of machine learning (ML) techniques into classical numerical solvers. Machine learning can enhance solver performance by **learning solution patterns**, **accelerating convergence**, and **reducing computational costs**, particularly for high-dimensional or nonlinear systems. Unlike traditional solvers that rely solely on discretization and linear algebra, ML-enhanced solvers leverage **data-driven insights** to approximate or accelerate solutions, enabling hybrid approaches that combine physics-based modeling with statistical learning.

Overview

ML integration in numerical solvers can generally be classified into three main approaches:

1. **Neural Network-based Solvers:** Deep neural networks (DNNs) are used to learn solution functions directly from datasets generated by classical solvers or experiments. The trained networks can then approximate solutions rapidly for new inputs.
2. **Physics-Informed Neural Networks (PINNs):** PINNs embed physical laws into the learning process. Instead of relying purely on labeled data, the network is constrained to satisfy governing differential equations, boundary conditions, and initial conditions, which ensures physical consistency.
3. **Surrogate Models:** Surrogate modeling involves replacing computationally expensive components of a solver with ML-based approximations. These models predict outcomes quickly, enabling fast parametric studies, optimization, and uncertainty quantification.

The selection of the appropriate ML integration approach depends on problem complexity, data availability, and required accuracy.

Neural Network-based Solvers

Neural networks are universal function approximators capable of modeling highly nonlinear relationships. In the context of numerical solvers, they learn a mapping between **input parameters** (e.g., boundary conditions, material properties) and **solution outputs** (e.g., stress fields, temperature distributions, velocity profiles).

Page 39-51 © MANTECH PUBLICATIONS 2021. All Rights Reserved

Workflow:

1. **Data Generation:** A dataset is collected by solving the governing equations using classical numerical solvers (FEM, FDM, etc.) for different parameter sets.
2. **Network Training:** A neural network is trained to minimize a loss function, typically the mean squared error (MSE) between predicted solutions u^(x)\hat{u}(x)u^(x) and ground truth solutions u(x)u(x)u(x):

L(θ)=1N∑i=1N|u^θ(xi)−u(xi)|2\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left| \hat{u}_\theta(x_i) - u(x_i) \right|^2L(θ)=N1i=1∑N|u^θ(xi)−u(xi)|2

where θ\thetaθ denotes network parameters, and NNN is the number of training samples.

3. **Deployment:** Once trained, the network can predict solutions for new input configurations almost instantaneously.

Advantages:

- Efficient for high-dimensional input spaces.
- Provides rapid predictions after training, significantly reducing computational time.
- Useful for real-time applications and parametric studies.

Limitations:

- Requires large, high-quality datasets for training.
- Generalization outside the training domain can be poor.
- Cannot guarantee strict adherence to physical laws without modifications.

Example Application:

In **structural mechanics**, neural networks have been trained to predict stress and strain distributions in complex geometries under varying loads. Once trained, the network bypasses repetitive FEM computations, providing real-time predictions during design optimization.

Physics-Informed Neural Networks (PINNs)

Physics-Informed Neural Networks (PINNs) represent a significant advancement in combining ML with numerical solvers. Unlike traditional neural networks, PINNs **incorporate the**

Page 39-51 © MANTECH PUBLICATIONS 2021. All Rights Reserved

governing physics directly into the loss function, allowing the model to learn solutions while strictly satisfying physical laws.

Formulation:

For a differential equation

N[u(x)]=f(x),x∈Ω\mathcal{N}[u(x)] = f(x), \quad x \in \OmegaN[u(x)]=f(x),x∈Ω

a PINN approximates u(x)u(x)u(x) as uθ(x)u_\theta(x)uθ(x) using a neural network and minimizes the residual:

L(θ)=1N∑i=1N|N[uθ(xi)]−f(xi)|2\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left| \mathcal{N}[u_\theta(x_i)] - f(x_i) \right|^2L(θ)=N1i=1∑N|N[uθ(xi)]−f(xi)|2

Boundary and initial conditions are incorporated as additional penalty terms:

LBC=1M∑j=1M|uθ(xj)−g(xj)|2\mathcal{L}_{BC} = \frac{1}{M} \sum_{j=1}^{M} \left| u_\theta(x_j) - g(x_j) \right|^2LBC=M1j=1∑M|uθ(xj)−g(xj)|2

where g(xj)g(x_j)g(xj) represents known boundary values.

Advantages:

- Reduces dependence on large labeled datasets.
- Ensures solutions are physically consistent by satisfying governing equations.
- Applicable to both forward problems (predicting states) and inverse problems (identifying parameters or source terms).
- Particularly effective for complex PDEs and high-dimensional problems.

Limitations:

- Training can be computationally intensive, sometimes comparable to traditional solvers.
- Convergence is sensitive to hyperparameters, neural network architecture, and training strategies.
- May require adaptive weighting between PDE residuals and boundary condition penalties.

Applications:

- **Fluid Dynamics:** Solving Navier–Stokes equations for turbulent or laminar flows without traditional grid-based simulations.

Page 39-51 © MANTECH PUBLICATIONS 2021. All Rights Reserved

- **Heat Transfer:** Predicting temperature distribution in heterogeneous or anisotropic materials.
- **Electromagnetic Simulations:** Solving Maxwell's equations for complex boundary conditions.

Example:

Raissi et al. (2019) demonstrated the use of PINNs to solve two-dimensional incompressible Navier–Stokes equations, accurately predicting velocity and pressure fields using minimal training data compared to conventional CFD solvers.

Surrogate Models

Surrogate models provide a **data-driven approximation** of a computationally expensive numerical solver. Instead of solving the governing equations directly, a surrogate predicts outputs based on prior evaluations of the system. Surrogate models are particularly useful in **optimization**, **uncertainty quantification**, and **sensitivity analysis**.

Common Surrogate Techniques:

- **Gaussian Process Regression (GPR):** Provides probabilistic predictions with uncertainty estimates.
- **Support Vector Machines (SVMs):** Efficient for high-dimensional regression problems.
- **Ensemble Methods:** Random forests or gradient-boosted trees can approximate nonlinear mappings robustly.

Advantages:

- Rapid evaluation for new input configurations.
- Reduces computational cost in iterative studies or optimization loops.
- Provides uncertainty estimates in some methods (e.g., Gaussian processes).

Limitations:

- Accuracy is highly dependent on the quality, coverage, and representativeness of training data.
- May require retraining if the parameter space changes significantly.
- Does not inherently guarantee adherence to physical laws unless combined with physics-informed approaches.

Page 39-51 © MANTECH PUBLICATIONS 2021. All Rights Reserved

Example Application:

In **aerodynamic optimization**, surrogate models have replaced repeated CFD simulations, enabling faster exploration of design parameters like airfoil shapes and wing geometries. Gaussian process surrogates can predict lift and drag coefficients within seconds, compared to hours for full simulations.

COMPARATIVE ANALYSIS OF ML-ENHANCED SOLVERS

Table 1: Comparative Analysis of ML-Enhanced Solvers

Solver Type	Accuracy	Speed	Data Requirement	Physical Consistency
Classical FDM/FEM	High	Moderate	None	High
Neural Network-based	Moderate-High	Very High	High	Low
PINNs	High	High	Moderate	Very High
Surrogate Models	Moderate	Very High	High	Moderate

Page 39-51 © MANTECH PUBLICATIONS 2021. All Rights Reserved

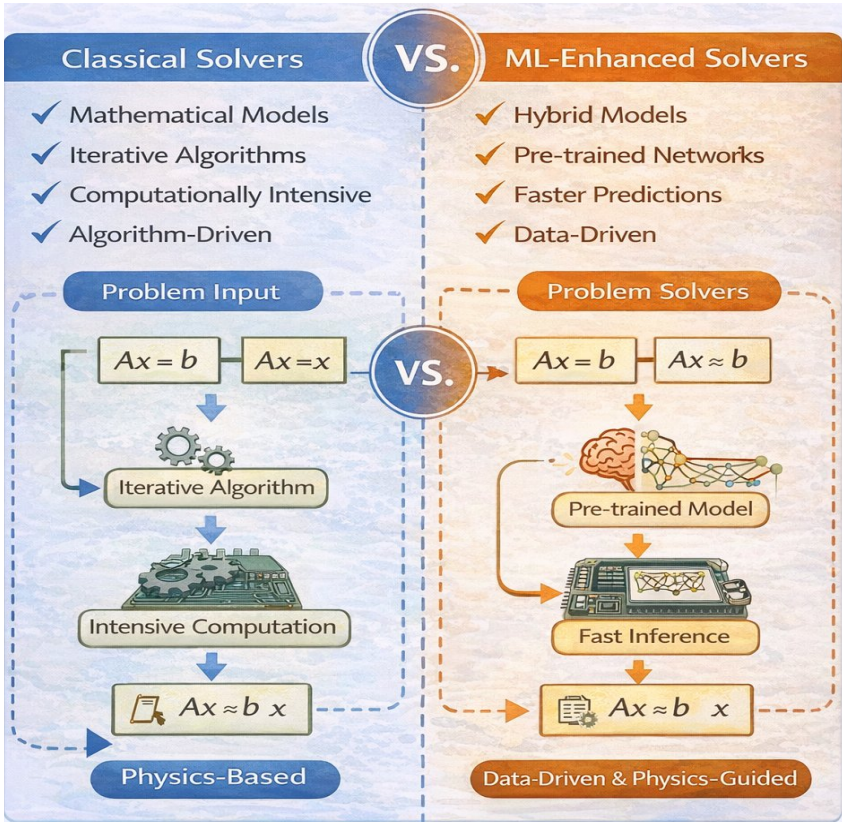


Figure 1: Schematic comparison of solver types (Classical vs. ML-enhanced)

Observations:

- ML-enhanced solvers excel in computational speed.
- PINNs uniquely balance accuracy and physical fidelity.
- Classical solvers remain robust for well-defined problems but scale poorly with system complexity.

APPLICATIONS OF MACHINE LEARNING ENHANCED SOLVERS

Computational Fluid Dynamics (CFD)

PINNs have been applied to model turbulent flows with reduced mesh requirements. Neural network solvers predict velocity and pressure fields, accelerating design cycles for aerospace and automotive applications.

Structural Analysis

ML-based solvers rapidly estimate stress and deformation in varying load conditions. Surrogate models allow engineers to perform parametric studies efficiently.

Electromagnetic Simulations

Page 39-51 © MANTECH PUBLICATIONS 2021. All Rights Reserved

High-frequency electromagnetic problems, traditionally solved using FEM or boundary element methods, benefit from ML-enhanced solvers for antenna design and wave propagation analysis.

Climate and Weather Modeling

High-dimensional PDEs in climate models are accelerated using surrogate models and PINNs, enabling faster scenario analysis and uncertainty quantification.

CHALLENGES AND FUTURE DIRECTIONS

Current Challenges

- **Data Availability:** Large, high-quality datasets are essential for neural network solvers.
- **Generalization:** ML models may fail outside their training or parameter ranges.
- **Computational Cost:** Training PINNs can be as expensive as classical solvers.
- **Interpretability:** ML models often behave as black boxes, complicating verification.

Future Research Directions

1. **Hybrid Solvers:** Combining classical and ML-based methods for adaptive, efficient simulations.
2. **Adaptive Learning:** Online learning methods that refine ML models during simulations.
3. **Uncertainty Quantification:** Integrating probabilistic ML models to assess solution confidence.
4. **Hardware Acceleration:** Leveraging GPUs, TPUs, and neuromorphic computing for faster training.
5. **Cross-disciplinary Applications:** Expanding ML-enhanced solvers to finance, medicine, and materials science.

CONCLUSION

Machine learning-enhanced numerical solvers represent a significant advancement in computational science, enabling faster and potentially more accurate solutions to complex problems. Techniques such as neural network-based solvers, physics-informed neural networks, and surrogate models offer a range of tools suitable for high-dimensional, nonlinear,

Page 39-51 © MANTECH PUBLICATIONS 2021. All Rights Reserved

and parametric systems. While challenges remain in terms of data requirements, generalization, and interpretability, the synergy between classical numerical methods and machine learning promises a transformative impact across scientific and engineering disciplines. Future research focusing on hybrid approaches, adaptive learning, and uncertainty quantification will further enhance the applicability and reliability of these solvers.

REFERENCES

1. Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics, 378*, 686–707.
2. Zhang, Y., & Chen, W. (2020). Machine learning-assisted numerical methods for differential equations: A review. *Applied Mathematical Modelling, 78*, 323–345.
3. Hesthaven, J. S., & Ubbiali, S. (2018). Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics, 363*, 55–78.
4. Ling, J., Kurzawski, A., & Templeton, J. (2016). Reynolds averaged turbulence modeling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics, 807*, 155–166.
5. Karniadakis, G. E., Kevrekidis, I., Lu, L., Perdikaris, P., & Wang, S. (2021). Physics-informed machine learning. *Nature Reviews Physics, 3*, 422–440.
6. Tripathy, R. K., & Bilionis, I. (2018). Deep UQ: Learning deep neural network surrogate models for high-dimensional uncertainty quantification. *Journal of Computational Physics, 375*, 565–588.
7. Raissi, M., Yazdani, A., & Karniadakis, G. E. (2020). Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science, 367*, 1026–1030.
8. Wang, S., Teng, Y., & Perdikaris, P. (2021). Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing, 43*(5), A3055–A3081.
9. Brunton, S. L., & Kutz, J. N. (2019). Data-driven science and engineering: Machine learning, dynamical systems, and control. Cambridge University Press.

Page 39-51 © MANTECH PUBLICATIONS 2021. All Rights Reserved

10. Duraisamy, K., Iaccarino, G., & Xiao, H. (2019). Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics, 51*, 357–377.

Page 39-51 © MANTECH PUBLICATIONS 2021. All Rights Reserved