International Journal of Engineering Research-Online
A Peer Reviewed International Journal
ISSN: 2321-7758 http://www.ijoer.in editorijoer@gmail.com

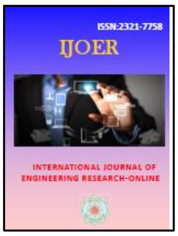
Vol.14., S1, 2026
January

Special issue

ISSN: 2321-7758

From Grids to Graphs: Hybrid Numerical Solvers for Differential and Integral Equations in CFD and Renewables

Thripurala Renuka Devi
Lecturer in Mathematics
Y.V.N.R. Government Degree College, Kaikaluru
Email:renukadevithripurala@gmail.com
DOI: 10.33329/ijoer.14.S1.22



Abstract

Differential and integral equations underpin models in physics, engineering, and computational fluid dynamics. Classical numerical methods, including finite differences, finite elements, and collocation schemes, discretize continuous operators with proven convergence but high computational demands for complex geometries or multiscale phenomena. Data-driven approaches, such as sparse regression (e.g., SINDy), symbolic regression via genetic programming, and deep learning (e.g., PINNs, neural operators), infer governing laws from simulation or observational data, enabling discovery of unknown dynamics and coarse-graining at reduced resolutions. This review categorizes methodologies, benchmarks performance on canonical equations like Burgers' and Navier-Stokes, discusses hybrids for enhanced robustness, and outlines challenges in uncertainty quantification and scalability. Hybrids promise interpretable, data-efficient solvers for real-world applications in renewable energy modeling and multiphase flows.

Keywords: Finite difference methods, Physics-informed neural networks, Galerkin methods, Sparse identification (SINDy), Neural operators.

Introduction

Differential equations (DEs) describe dynamic evolution in systems from fluid flow to quantum mechanics, while integral equations (IEs) model nonlocal interactions like potential theory. Ordinary DEs (ODEs) govern time-dependent processes, partial DEs (PDEs) incorporate space, and IEs arise in Fredholm or Volterra forms for inverse problems. Numerical solution demands balancing accuracy, stability, and efficiency, especially for nonlinear stiff systems prevalent in computational fluid dynamics (CFD) and renewable energy simulations [1,2].

Classical methods discretize domains into grids or mesh, approximating derivatives via Taylor expansions or weak formulations. Finite difference methods (FDMs) suit regular grids, finite element methods (FEMs) irregular domains, and spectral methods smooth solutions. For IEs, Nyström quadrature or projection methods reduce to DEs. These yield systems solvable by explicit/implicit time-stepping like Runge-Kutta or backward Euler, with error bounds via Lax equivalence theorem:

consistency plus stability implies convergence [3].

Data-driven methods emerged post-2016, fueled by AI advances. Sparse regression builds candidate libraries (polynomials, trig functions) and prunes via L1 penalties, discovering sparse nonlinear laws (SINDy). Symbolic regression evolves expressions genetically. Neural networks parameterize solutions (Neural ODEs) or operators (FNO), embedding PDE residuals in losses for mesh-free solving. Data-driven discretization learns stencil coefficients from high-res data, achieving 4-8x coarser grids without divergence [2].

This review structures as: Methodology details classical and data-driven solvers; Discussion compares via benchmarks, hybrids; Conclusion charts paths forward. Emphasis lies on CFD relevance, aligning with multiphase flow and sustainability modelling [4].

Methodology

Classical Approaches

Finite Difference and Finite Volume Methods

FDMs approximate derivatives: central $\frac{u_{i+1}-u_{i-1}}{2\Delta x}$ for first-order, higher via stencils. For hyperbolic PDEs like advection $u_t + au_x = 0$, upwind schemes ensure CFL stability $| a | \Delta t/\Delta x \leq 1$. Finite volume conserves fluxes: $\frac{d}{dt}\bar{u}_i = \frac{F_{i-1/2}-F_{i+1/2}}{\Delta x}$, with Riemann solvers for shocks (Godunov). WENO adaptively weights smooth stencils, suppressing oscillations.[5]

For viscous Burgers' $v_t + (v^2/2 - \eta v_x)_x = f$, FVM integrates cell averages, resolving shocks at width $\sim \eta$. Navier-Stokes extends via projection methods decoupling velocity-pressure [2].

Finite Element and Galerkin Methods

FEMs minimize residuals in weak form: $\int \nabla u \cdot \nabla v = \int fv$, yielding sparse matrices solved by CG. For IEs, Galerkin projects kernels onto bases, e.g., discrete $\sum \phi_j(x)K(x,y)u(y)dy = f(x)$. hp-adaptive refines elements [6].

Time integration: explicit for non-stiff, implicit (BDF) for stiff, multigrid acceleration [2].

Integral Equation Solvers

Volterra IEs use trapezoidal quadrature; Fredholm via Nystrom $\sum w_k K(x_k, y_k)u(y_k) = f(x_i)$. Boundary IEs reduce dimensionality [7].

Data-Driven Approaches

Sparse and Symbolic Regression

SINDy forms $\dot{u} = F\Xi$, libraries $F = [1, u, u^2, sin\ u, \ldots]$, solves $\Xi = arg\ min \parallel \dot{U} - F\Xi \parallel_2^2 + \lambda \parallel \Xi \parallel_1$. Derivatives via TVD or splines handle noise. STRidge iterates thresholding+ridge for PDEs. Bayesian variants (UQ-SINDy) use spike-slab priors for inclusion probabilities [8].

Symbolic regression genetically evolves trees: fitness MSE on integrated predictions. Eureqa partitions variables, probes ICs [7].

Neural Network-Based Methods

PINNs minimize $L = \parallel u - u^* \parallel^2 + \parallel N[u] \parallel^2$, where N is PDE residual, derivatives via autodiff. Solves forward/backward without mesh [9].

Neural ODEs $\frac{du}{dt} = f_\theta(u, t)$, integrate via adjoint sensitivity. FNO learns $G(a) = u$ via Fourier transforms, mesh-invariant [10].

Data-driven discretization: NN predicts stencils $\partial^n v/\partial x^n = \sum \alpha_i^n v_i$, optimized for time deriv accuracy on coarse grids from fine data. Pseudolinear ensures polynomial order [11].

Category	Technique	Library/Arch	Strengths	Comp. Cost
Classical	FDM/WENO	Fixed stencils	Stability proofs	O(N log N)
Classical	FEM/Galerkin	Basis funcs	Adaptive meshes	$O(N^{\{3/2\}})$
Sparse	SINDy/SR3	Polynomials	Interpretable eqs	Low
Neural	PINNs	MLP	Mesh-free	Train: High, Infer: Low
Operator	FNO	Fourier NN	Resolution indep	Medium

Hybrids: NN-denoised derivatives fed to SINDy; PSM Bayesian hierarchies [7].

Discussion

Benchmarks and Comparisons

On Burgers' (viscous shock), classical FVM/WENO diverges at 16x coarse (shock width unresolved), data-driven NN integrates stably, MAE 8x lower (Fig. 3C page:1). KdV solitons, KS chaos: NN valid sim time 2-5x longer at 8x coarse.

Lorenz ODEs: SINDy recovers $\dot{x} = \sigma(y - x)$ from noisy data, Bayesian UQ-SINDy quantifies 95% CIs. Navier-Stokes: PINNs handle no-slip BCs, FNO parametric Re [12].

IEs: Galerkin exact on smooth kernels; NN operators generalize [13]. Classical excel interpretability/stability; data-driven accuracy/data scarcity (e.g., CFD sims costly, but few suffice training).

Advantages and Limitations

Classical: Mesh gen, preconditioners needed; scale poorly d>2 .

Data-driven: Black-box risks, train data req (10^4-10^6 snapshots), overfitting. Noise sensitivity mitigated by GP priors, ensembles .

Uncertainty: Bootstrap/SR3 for param UQ; full Bayesian PSM propagate obs-process errors [7].

CFD Apps: Multiphase flows (VOF+level-set), renewables (wind turbine wakes): hybrids NN-subgrid + classical solver [4].

Scalability: FNO O(1) eval post-train; quantum hybrids for exp large [14].

Hybrids and Emerging Trends

SR3+NN: NN approx u, sparse on derivs. Operator inference learns low-rank; DLGA-PDE genetic+NN for param PDEs [7]. Quantum-classical solvers for linear DEs. Green AI: low-data symbolic [14]. Challenges: Provable guarantees (neural Tangent Kernel), equivariance, multi-fidelity.

Metric	Classical	Data-Driven	Hybrid
Accuracy (coarse)	Low	High	Highest
Interpretability	High	Low	Medium
Data Req	None	High	Medium
UQ	Analytic	Ensemble	Bayesian

Conclusion

Classical methods furnish bedrock for DE/IE solvers; data-driven unlock coarse, data-sparse regimes. Hybrids, uncertainty-aware, herald CFD revolutions in renewables. Future: certified hybrids, 3D operators, physics-equivariant nets.

References

[1]. Collins, G. W., II. (1990). *Fundamental numerical methods and data analysis* (Chapter 5). Harvard Astrophysics Data System. https://ads.harvard.edu/books/1990fnmd.book/chapt5.pdf.

[2]. "Advances in Numerical Methods for Differential and Integral Equations." *Contemporary Mathematics*, Wiserpub, 30 Dec. 2024, ojs.wiserpub.com/index.php/CM/SI/Num_Differ_integral.

[3]. Cameron, M. (n.d.). *Numerical methods for solving ordinary differential equations* [Lecture notes]. University of Maryland, Department of Mathematics. https://www.math.umd.edu/~mariakc/AMSC661/LectureNotes/ODEsolvers.pdf

[4]. Bar-Sinai, Y., Hoyer, S., Hickey, J., & Brenner, M. P. (2019). Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31), 15344-15349. https://doi.org/10.1073/pnas.1814058116.

[5]. "Finite difference method." *Wikipedia, The Free Encyclopedia*. Wikimedia Foundation. Retrieved January 8, 2026, from https://en.wikipedia.org/wiki/Finite_difference_method.

[6]. Atkinson, K. E. (1988). A discrete Galerkin method for first kind integral equations with a logarithmic kernel. *Journal of Integral Equations and Applications*, 1(3), 343-363. https://doi.org/10.1216/JIE-1988-1-3-343.

[7]. Martinsson, P.-G. (2019). *Integral equation formulations* [Lecture notes]. University of Texas at Austin, Oden Institute. https://users.oden.utexas.edu/~pgm/Teaching/2019_393C/classic.pdf.

[8]. North, J. S., Wikle, C. K., & Schliep, E. M. (2022). *A review of data-driven discovery for dynamic systems* (arXiv:2210.10663v1 [stat.ME]). arXiv. https://arxiv.org/abs/2210.10663.

[9]. Matthews, J., & Bihlo, A. (2024). *PinnDE: Physics-informed neural networks for solving differential equations* (arXiv:2408.10011v1 [cs.LG]). arXiv. https://arxiv.org/abs/2408.10011.

[10]. Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2021). *Fourier neural operator for parametric partial differential equations* (arXiv:2010.08895v3 [cs.LG]). arXiv. https://arxiv.org/abs/2010.08895.

[11]. Bar-Sinai, Y., Hoyer, S., Hickey, J., & Brenner, M. P. (2019). Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(45), 22405-22410. https://doi.org/10.1073/pnas.1814058116.

[12]. "Local neural operator for solving transient partial differential equations on varied domains" *Computer Methods in Applied Mechanics and Engineering*, Volume 427, 1 July 2024, 117062.

[13]. https://academic.oup.com/imajna/article-abstract/5/3/355/880178?__cf_chl_tk=A8s3IqciI2pvhaAbm8DAvLnGV2lRxKpXEGm.nbEdZKw-1767839060-1.0.1.1-vYp6bqcuGu_pVqd.vq0QeD0tU97H.yjA6rUNuUK6AA8.

[14]. Jaffali, H., de Araujo, J. B., Milazzo, N., Reina, M., de Boutray, H., Baumann, K., Holweck, F., Mohdeb, Y., & Katz, R. (2025). *H-DES: A quantum-classical hybrid differential equation solver* (arXiv:2410.01130v2 [quant-ph]).