 © Authors of the article / Authors of the article

МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ, КОМПЛЕКСОВ И КОМПЬЮТЕРНЫХ СЕТЕЙ/MATHEMATICAL SOFTWARE FOR COMPUTERS, COMPLEXES AND COMPUTER NETWORKS

DOI: https://doi.org/10.60797/itech.2026.11.2 EDN: RODTYQ

A SYSTEMATIC REVIEW OF MODERN NUMERICAL METHODS FOR SOLVING PARTIAL DIFFERENTIAL EQUATIONS

Review article

Saba S.M.[1, *]

[1] University of Baghdad, Baghdad, Iraq

* Corresponding author (saba.s[at]coeng.uobaghdad.edu.iq)

Suggested: 04.02.2026; Accepted: 14.04.2026; Published: 14.07.2026

Abstract

This article is a systematical review on the current numerical methods used in the numerical solution of partial differential equations (PDEs) in which our special attention is focused on the latest developments essentially around the past decade. It starts to summarize standard classifications of PDE, and briefly returns to the classical classification of discretization mixed methods, mainly finite difference, finite element, or finite volume, which can get a preliminary knowledge in order to make a scientific comparison.

The review then gives an overview of more recent developments of spectral and spectral element methods, meshfree and material point methods, adaptive and multiscale formulations, and multigrid-based acceleration methods. One of the major contributions of the present article is the learning-based and probabilistic paradigm. In particular, the manuscript talks about AI-driven approaches such as physics informed neural networks that help the operator learn using DeepONet, Fourier neural operators and other hybrid approaches that combine data-driven factors and classical numerical solvers for operator learning.

Across these methodological families, the article makes a qualitative comparison in terms of aspects of accuracy, computational cost, stability, scalability, implementation complexity, and representative application domains. Illustrative examples cover the fields of fluid dynamics, solid mechanics, biomedical flow modelling, as well as problems arising in the problems of finance and quantum systems.

The review concludes with an open questionnaire, in particular the need for more drastic theoretical guarantees and findings on uncertainty quantification combined with the need for efficient coupling with high-performance computing environments. It describes promising avenues for future research, in particular in rigorously analyzed hybrid approaches and solvers based on operator learning.

Keywords: partial differential equations, numerical methods, spectral methods, finite element method, meshfree methods.

СИСТЕМАТИЧЕСКИЙ ОБЗОР СОВРЕМЕННЫХ ЧИСЛЕННЫХ МЕТОДОВ РЕШЕНИЯ УРАВНЕНИЙ В ЧАСТНЫХ ПРОИЗВОДНЫХ

Обзор

Саба С.М.[1, *]

[1] Багдадский университет, Багдад, Ирак

* Корреспондирующий автор (saba.s[at]coeng.uobaghdad.edu.iq)

Предложена: 04.02.2026; Принята: 14.04.2026; Опубликована: 14.07.2026

Аннотация

Это статья представляет собой систематический обзор современных численных методов, применяемых для численного решения уравнений в частных производных (УЧП), при этом особое внимание уделяется новейшим достижениям, появившимся главным образом за последнее десятилетие. Сначала в ней излагаются стандартные классификации УЧП и кратко рассматривается классическая классификация методов дискретизации (в основном разностные, конечно-элементные и конечно-объёмные методы), что позволяет получить первоначальное представление и провести научное сравнение.

Затем в обзоре даётся характеристика более современных направлений: спектральных и спектрально-элементных методов, бессеточных методов и метода материальной точки, адаптивных и многомасштабных постановок, а также методов ускорения на основе многосеточных алгоритмов. Одним из основных вкладов данной статьи является рассмотрение обучаемой и вероятностной парадигмы. В частности, рукопись обсуждает подходы, основанные на искусственном интеллекте, такие как physics-informed neural networks, а также методы обучения операторов, включая DeepONet, Fourier Neural Operator и другие гибридные подходы, сочетающие управляемые данными компоненты с классическими численными решателями для операторного обучения.

Для всех этих методологических семейств в статье проводится качественное сравнение по таким аспектам, как точность, вычислительная стоимость, устойчивость, масштабируемость, сложность реализации и типичные области применения. Иллюстративные примеры охватывают задачи гидродинамики, механики твёрдого тела, биомедицинского моделирования течений, а также задачи, возникающие в финансах и квантовых системах.

 © Authors of the article / Authors of the article

В заключение обзор формулирует ряд открытых вопросов, в частности необходимость более строгих теоретических гарантий и результатов в области количественной оценки неопределённости, а также потребность в эффективной интеграции с высокопроизводительными вычислительными средами. Описываются и перспективные направления будущих исследований, особенно в области строго обоснованных гибридных подходов и решателей, основанных на обучении операторов.

Ключевые слова: уравнения в частных производных, численные методы, спектральные методы, метод конечных элементов, безсеточные методы.

Introduction

Partial differential equations (PDEs) form the very fundamental building blocks of mathematical modeling in a very wide range of domains of science and engineering, such as fluid dynamics, heat and mass transfer, elasticity, electromagnetics, as well as quantitative finance, where they describe the spatiotemporal evolution of physical fields [1].

During the last decades, classical discretization techniques, among others, finite difference, finite element and finite volume techniques, have enabled reliable large-scale computations in a number of application domains. However, these well-established methodologies are challenging to employ in solving complex problems that have complex geometries, highly nonlinear dynamics, and large parametric dimensionality, in which issues related to stability, the level of accuracy and calculation cost become more and more pronounced [12].

All these limitations have allowed the current interest in modern numerical methods that aim to improve the transfer of robustness and predictive quality while reducing the computational effort, especially through better scalability on parallel architectures and the introduction of data-driven components. Against this background, the current review is aimed to systematically review recent developments in solvers of partial differential equations reported in the period 2015 to 2025, by focusing particular attention on spectral and meshfree formulations, adaptive and multiscale approaches, artificial intelligence-based and probabilistic solvers, as well as advancing hybrid approaches, where learning modules are combined with standard numerical schemes. The manuscript outlines at first the required mathematical and numerical prerequisites before giving a brief historical overview of the main ideas of the major contemporary methodological families. This is followed by an evaluative appraisal of their respective strengths and trade-offs in terms of the key performance criteria. Representative applications are highlighted throughout the discourse, and outstanding challenges and possible avenues for future research are stated in the conclusion of the review.

Classical background

Simple classification and intrinsic properties of problems of partial differential equations. Typically, Partial Differential Equations fall into one of three groups — elliptic, parabolic, and hyperbolic — depending on the algebraic structure of the highest order term of the differential operator, and it is directly related to the qualitative nature of its solutions [1].

In the problems of electrostatics, elastic theory and potential flow dynamics, elliptic equations like the Laplace and Poisson equations are used widely. Most often these solutions are smooth, meaning that the boundary conditions affect the solution in and around the region. For example, parabolic equations, such as the heat equation, and the Fokker-Planck equation are of interest, since they model diffusive, dissipative or relaxation phenomena; in such systems, the solution evolves with time, and, by necessity, "smoothes" the irregularities, and approaches an equilibrium state, while, at the same time, remaining constrained by the set of initial data and boundary conditions. This is the control of finite-slope propagation of signals and characteristics defined by hyperbolic equations like the wave and advection equations. Their solutions may show wave-like behavior and even exhibit steep gradients and propagating fronts, or discontinuities, particularly for nonlinear regimes or more complicated forcing. From the modelling point of view, a problem of the above type is said to be well — posed if it has the properties: existence, uniqueness, and dependence on the initial and/or boundary data that is continuous with respect to those data. Well, poshness is essential for mathematical soundness as well as for numerical reliability: discretization schemes, which allegedly fail to meet these criteria, e.g., due to incompatible treatment of the boundary or inadequate stability control, can provide relevant unstable, divergent or physically implausible solutions [1]. In general, there are many models that are used in practice where the methods of the classical strong formulation become too restrictive, and then a weak or variational formulation in suitable Sobolev spaces is used [1].

For a representative **linear elliptic problem**

$$-\nabla \cdot (a\nabla u) = f \qquad \text{in } \Omega \tag{1}$$

The weak formulation can be expressed as:

$$\int_\Omega a\nabla u \cdot \nabla v dx = \int_\Omega f v dx + \int_{\partial\Omega} v(a\nabla u \cdot n) ds \tag{2}$$

The derivatives naturally induce moving from the unknown to the test function, the discretization's of the above partial differential equations in Galerkin form on which the finite element family of methods [12] is based.

Basic schemes for number crunching Foundational numerical methods to partial differential equations. Foundational method for numerical solution of partial differential equations (PDEs): systematic discretization of the space-time domains for computing massive algebraic systems that approximate the continuous problems. The finite difference method (FDM) involves computing difference quotients of spatial derivatives at points in a structured grid in simple stencil operations and easy to implement on regular domains [12]. The most important strengths of this are its conceptual beauty and the low computation cost it has when implemented on rectangular or logically Cartesian lattices with high order/ compact finite — difference stencils. However, it is important to realize that when applied to domains of intrinsically complex geometry, the approach may

 © Authors of the article / Authors of the article

become unwieldy, and its freedom of choice of boundary conditions beyond the, rather conventional, boundary conditions is, unfortunately, somewhat limited.

In contrast, the FEM relies on a variational or weak formulation, and represents the solution by a patchwork of piecewise polynomial basis functions defined on a system of elements which need not be arranged in any special way and can be quite unstructured. First, such methodology issues as convergence, stability and the intrinsic complexity of the problem can be dealt with in a strict manner; second, complex geometries and different materials can be accommodated; and third, a vast array of boundary conditions can be imposed in a consistent and natural manner. In the context of finite element theory, discontinuous Galerkin (DG) methods introduce discontinuous basis functions at interfaces between elements and use numerical fluxes on interfaces to couple the elements together [2]. DG methods offer not only a high degree of accuracy, but also local conservation, favorable operator locality — both highly desirable for parallel implementations as well as for hp-adaptive refinement strategies. It is true that in general these amazing benefits are associated with higher computational complexity and memory consumption per degree of freedom when compared with conventional continuous FEM formulations.

The finite volume method, on the other hand, is conservation-based, where the governing equations are put into the control volumes, and the conservation is applied through a balance of fluxes at the faces of the cells [12]. The property of conservation inherent in the finite volume approach and its capability to deal with the structured and unstructured mesh has now made FVM a very famous method in the congress of computational fluid dynamics and transport modeling. However, to obtain a high level of accuracy and robustness (especially if one wants to implement one of these strategies with complex source terms, steep gradients or nonlinear fluxes) more complex reconstruction techniques, carefully designed limiters, and reliable Riemann solvers for the fluxes to be calculated are often required.

The classic finite difference, finite element (including discontinuous Galerkin) and finite volume methods are collectively referred to as finite difference type schemes, and new schemes using spectral, mesh-free, adaptive and AI-assisted methods are evaluated against them, particularly for accuracy, robustness and computational efficiency.

Modern numerical methods

Spectral and spectral element methods approximate solutions of partial differential equations by expanding the unknown field as an expansion in globally supported basis functions, usually trigonometric polynomials (e.g. Fourier bases) and families of orthogonal polynomials, and solving for the corresponding expansion coefficients from the governing equations and accompanying constraints [2]. Spectral methods approximate the solution $u(x)$ as a series expansion of smooth and globally supported basis functions $\phi_k(x)$:

$$u(x) \approx u_N(x) = \sum_{k=0}^{N} u_k \phi_k(x) \tag{3}$$

where N is the number of degrees of freedom, and $u^k \phi_k(x)$ are the spectral coefficients.

The spectral collocation (or pseudo-spectral) method is the most widely used approach. The PDE is enforced exactly at a set of collocation points, such as Chebyshev Gauss Lobato points. If the PDE is given by $Lu = f$ where L is a differential operator, then the collocation method imposes:

$$Lu_N\left(x_j\right) = f\left(x_j\right), \quad orj = 0, 1, \ldots, N \tag{4}$$

Where x_j are the collocation points. The derivatives of u_N are computed using spectral differentiation matrices, which convert the differential operator L into a matrix D . The PDE is then transformed into a linear algebraic system:

$$\boldsymbol{D}u_N = f \tag{5}$$

Spectral schemes in their classical global form can converge very rapidly, with a high degree of smoothness of the solution they represent; many spectral schemes can be exponentially convergent with the number of modes retained. This is due to the fact that typically a small number of modes can represent the large-scale features of the solution with great fidelity [2]. Simultaneously, the global support of the basis functions makes them impractical. Spectral methods are more sensitive to the loss of regularity, for example, no limits on smoothness may lead to Gibbs-type oscillations, which affect the accuracy of the solution. The root cause of this is the global coupling present in these techniques, which complicates even further the numerical resolution of extremely localized phenomena. In addition, the global basis functions make it rather tedious to account for complex geometries and non-standard boundaries, as compared to formulations with locally supported discretisation's [2].

By merging the spectral accuracy and element-wise domain decomposition, the spectral-elements approach overcomes some of the above drawbacks. The computational domain is subdivided into elements, and the solution is approximated within each element by high order polynomial functions defined on a reference element which gets mapped to the physical element [2]. The high-order accuracy that is associated with global spectral methods is largely retained, and the geometry adaptability of the traditional finite element methods proved very useful in the cases that we consider for which simulation required the handling of complex domains. For this reason, spectral and spectral-element methods have been extensively applied to various problems such as fluid flow, wave propagation and structural dynamics, particularly when the solution is fairly regular and high-performance computing and use of efficient solvers are pertinent.

In representing the solution field, meshfree methods do not involve any fixed connectivity mesh, but rather a set of randomly placed nodes or particles, from which the solution field is reconstructed by using a set of suitable shape functions or kernel-based interpolants [9]. Since this element topology is not specified explicitly, the above approaches offer a simplified discretization where the problem of the mesh generation or remeshing, or increasing the quality of the mesh, would be a dominating issue.

 © Authors of the article / Authors of the article

For instance, radial basis function (RBF) methods approximate the solution using a superposition of radially symmetric basis-functions that are centered at the nodes and that can be used to achieve high-order accuracy, even for irregular node-distributions, and are relatively easy to be extended to higher dimensions [3].

In RBF methods, the solution $u(x)$ is approximated as a linear combination of radial basis functions Φ, which depend only on the radial distance $r = \|x - x_i\|$ between the evaluation point x and the node x_i :

$$u(x) \approx \sum_{t=1}^{N} \lambda_t \Phi\left(\|x - x_t\|\right) + P(x) \tag{6}$$

Where λ_i are the unknown coefficients, and $P(X)$ is a polynomial term added to ensure accuracy and stability.

Smoothed particle hydrodynamics (SPH) adopts a fully Lagrangian description in which the continuum is represented by moving particles that carry mass, momentum, and other state variables. Spatial derivatives are approximated through smoothing kernels, making SPH well-suited to large deformations, free-surface flows, and fragmentation phenomena that are challenging for mesh-based Eulerian discretization's [9]. The material point method (MPM) is a mixture of Lagrangian material points coupled to an Eulerian background grid. The grid is used to calculate the gradients, and the history dependent state information is carried with the particles, and the internal/external forces are calculated. This separation decreases the tangling of meshes and too much distortion of very severe distortion problems [10]. Methods that are meshfree in particular have been attractive for applications such as impact, fracture, granular flows and fluid-structure interaction since the repeated remeshing becomes overly complex or expensive [9]. Nevertheless, these benefits present tremendous difficulties. Numerous meshfree schemes are costly in terms of computational effort due to the construction and evaluation of shape functions or neighborhood searches and correct fulfillment of fundamental (Dirichlet-type) boundary conditions is nontrivial. Furthermore, the solution stability (tensile instability, some types of particle formulation) should be taken into account, and some stabilization measures should be followed [9].

Adaptive, MultiGrid and Multiscale Methods. Adaptive approaches aim to allocate the available computational effort in those regions, which can provide the most information from existing or required detailed spatial or temporal discretization in regions of high or estimated error. This kind of principle can be realized through the technique of adaptive mesh refinement (AMR), refining and coarsening the computational mesh according to the indications of errors or error estimators that can improve the efficiency of computing the localized features like shocks, boundary layers and sharp interface without any uniform refinement of the entire region [7]. To enhance the computational efficiency, the multigrid methods to accelerate the convergence of the iterative solvers are used. In the case of fine mesh computations, a combination of smoothing operations is used, and on the coarser mesh correction steps are used, thus allowing slow, low-frequency error components more easily to be removed. Many of these types of multilevel methods are near the complexity of almost optimal linear and nonlinear solution methods for large systems. Algebraic multigrid (AMG) offers more of these ideas: instead of visualizing the approximation process and employing geometric coarsening, the hierarchy is constructed directly from the discrete system matrix, i.e., AMG is naturally suitable for unstructured meshes and problems on complex geometries [8].

Multiscale methods and reduced-order models (ROMs) tackle problems, where phenomena on vastly different scales interact, or where repeated solutions of similar PDE's are required, for instance, in optimization and control problems. Variational multiscale formulations break up the solution into resolved and unresolved parts, which explicitly models the effects of the small-scale dynamics on the large-scale behavior so as to provide a systematic framework for the modelling of turbulence and subgrid scale processes. ROM techniques, e.g., based on proper orthogonal decomposition and on projection methods, give rise to low-dimensional dynamical systems that can be used to approximate the original PDE model with significantly reduced computational cost, but at the price of an offline training or basis construction phase.

Probabilistic and AI-based methods. One of the most recent developments is the use of machine learning combined with numerical analysis to create a new family of PDE solvers based on data-driven models subject to physical constraints [11]. Physics-informed neural networks (PINNs) incorporate all components of the problem, including the PDE, the boundary conditions, and the initial conditions into the loss function of a neural network to train the network to minimize the PDE residual, not labelled solution data, as was previously the case [4]. Each is equipped with a PINN, aiding in the localization of learning the high-frequency features.

Total Loss Function:

$$L(\theta) = \lambda_f L_f + \lambda_{bc} L_{bc} + \lambda_{ic} L_{ic} \tag{7}$$

where λ are weighting coefficients.

PDE Residual Loss $\left(L_f\right)$ **:** Computed based on the PDE residual f evaluated at random collocation points within the domain Ω .

$$L_f = \frac{1}{N_f} \Sigma \left\| f\left(x_i, t_i, u_\theta, \nabla u_\theta, \nabla^2 u_\theta\right) \right\|^2 \tag{8}$$

Derivatives are obtained via automatic differentiation.

Boundary and Initial Condition Losses $\left(L_{bc} \text{ and } L_{tc}\right)$ **:** Computed on boundary and initial points.

$$L_{bc} = \frac{1}{N_{bc}} \sum \left\| B\left(u_\theta\left(x_j, t_j\right)\right) - g\left(x_j, t_j\right) \right\|^2 \tag{9}$$

The network can automatically differentiate with spatial and temporal variables [4] and this paradigm is particularly intriguing for problems with high dimensions, inverse problems with few measurements, and cases where the data of the problem are costly to acquire. Operator learning methods, like Deep Operator Networks (DONs) [5] or Fourier Neural Operators (FNOs) [6] are designed to approximate the input to output mapping of an input function (or input), e.g., initial

 © Authors of the article / Authors of the article

conditions or source terms, to the resulting solution fields. These models can, once trained, give approximate answers for new inputs in more or less real time, and thus have enormous promise as surrogates for more cumbersome (and computationally demanding) solvers and models. Variants of these ideas, such as probabilistic variants (e.g., Gaussian process variants of the notion of physics-informed neural networks (PINNs)) and Bayesian variants of physics-informed neural networks (PINNs) further refine these concepts and enable principled uncertainty quantification to be generated in addition to the single-point estimate. These abilities can be particularly useful in risk-lovers decision-making, and in the realm of scientific inference, where analysis of confidence, and propagation of uncertainty, may be more important than the determination of a precise mean prediction [11]. However, following such conceptual elegance, there are significant limitations when it comes to practicality and theory in using data-driven solvers of PDEs. Typical problems are extremely high training expenses, dependence on (hyper) network topology and parameters, and stability and conservation laws (among others), which are easily guaranteed by classical discretization. Furthermore, there are only a few complete theoretical error bounds, the type of which would make it difficult to rigorously evaluate and reliably use in high-stakes scenarios many of the learning-based approaches.

Hybrid strategies try to combine complementary paradigms in a manner where each paradigm's strengths complement each other and ameliorate the drawbacks of each other. One approach frequently used is to discretize the various parts of the domain differently, for instance, using spectral or high-order finite element methods in smooth regions and finite volume or discontinuous Galerkin (DG) methods in regions with discontinuities or steep gradients or in regions with very complicated geometry, where robustness and conservation are more important [2]. Another emerging field of research is to embed classical solvers into the machine learning framework. Examples of this are neural models that predict the effective preconditioner, closure relations or SGS contribution that will allow a better convergence rate or physical fidelity in an approximately time-free fashion without sacrificing the underlying PDE discretization infrastructure [11]. Hybrid schemes similar to this, built on PINNs and traditional schemes are also emerging. By using high resolution simulations based on finite difference, finite element solvers or finite volume solvers a PINN can provide a source of training data, and the physics-based residual constraints in the PINN can be selectively activated with respect to subdomains of the parameter space (or the subsets of the parameter space). These couplings could reduce training overhead and even help to boost generalization and foster a de facto interoperability of existing and legacy codes with developed data-oriented tools.

All in all, hybrid and coupled methods are a promising route towards developing effective, efficient and flexible PDE solution strategies that can make use of the well-established techniques of classical numerical analysis and the growing power of current methods based on machine learning.

Comparative analysis and applications

Qualitative comparison. As far as accuracy is concerned, the spectral and spectral element methods converging with the increase of the resolution, are usually the fastest, especially if the solution is sufficiently smooth, and almost exponential convergence is obtained [2]. This performance improvement is explicitly pointed out in the literature on numerical approximation, and has to do with the global support and high regularity of the underlying basis functions. However, in a series of applications of partial differential equations (PDE) high-order finite-element formulations and discontinuous-Galerkin (DG) schemes can often provide a more reliable route to the systematic improvement of accuracy, given their very good ability to adapt systematically to complex geometries and/or localise high gradients [12].

Although methods such as meshfree methods and today's trading adaptive methods can be quite good at dealing with moving fronts, evolving interfaces, and sharp layers, their global error behavior, however, can be much more problem dependent and is less well known to characterize and study rigorously, and the stability and approximation properties can be sensitive to the distribution of nodes, adaptivity criteria, and choices of stabilization [3], [9].

Meanwhile, on many benchmarks, the accuracy of AI-based solvers can be the same or comparable to that of their classical counterparts; and on the academic side, it is well to note that the error capacity of these solvers is often not the only source of error, but also the data-covering and the dynamics of the operators trying to get to production, and the inductive biases of the architecture.

The trade-offs are no less nuanced, when it comes to the cost of computation. Even with well-matured linear and nonlinear solvers, e.g., multigrid and/or Krylov-subspace [12] methods, the conventional finite difference, finite element and finite volume methods remain amazing in terms of the efficiency per unknown. Such efficiency is largely owing to the very sparse structure of the operator and the well-known memory needs and decades of development both in terms of pre-conditioning as well as parallel implementation.

In contrast, meshfree methods and discontinuous Galerkin approaches have higher assembly and memory costs because of the higher complexity of the stencils, larger number of local degrees of freedom and the price of the construction of the shape functions or flux couplings [3], [9].

Spectral methods can be very competitive on simple geometries, where the rapidity of the transforms (using structured operators which do not incur a large overhead) can be very high, but when confronted with complex geometries of complicated boundaries, strong localisation and/or coefficients, this efficiency may drop drastically. In such an environment the advantages of the global representations are reduced and a higher degree of algorithmic complexity is needed [2].

Learning-based approaches have a specific cost model: They are likely to have high initial training costs on customized hardware but are cheap to evaluate once they have been trained. This "train one, evaluate many" paradigm is not only pivotal, but also involves a completely new way of thinking, for example, whether the training costs have to be taken into account, data and bias in the data, or mismatch between the different deployment domains if the methods are compared to classical solvers in a fair analytical framework [11].

In terms of stability and robustness, well-developed numerical schemes are numerically efficient due to the existence of a sound theoretical background, algorithmic constraints (e.g. CFL-conditions) and limiting and stabilization methods of

 © Authors of the article / Authors of the article

nonsmoothed solutions and systematic error estimation schemes. The reliability of these methods hinges on these attributes and they should have predictable and controllable failure modes in challenging engineering applications [12].

In contrast to this, there are not often extensive stability theories of meshfree techniques and the corresponding solvers based on artificial neural networks. They might have non-physical oscillations, which is a problem even in mechanics, they might lose their conservation properties or have some pathologies induced by the optimization which are not easy to pre-assess in the first place: thus, a long-lasting gap between empirical performance and serious guarantees [9] might exist.

Yet axes of scalability and implementation complexities still make a difference in practice when evaluating such approaches. The use of local formulations such as DG, adaptive mesh refinement (AMR) pipelines, and many meshfree particle methods is inherently conducive to parallelization, due to the fact that only localized evaluations of operators and limited communication patterns are used [7], [8]. Concurrently, using advanced adaptivity, multi-scale closures, and Al features in large production codes can significantly enhance the code complexity, the work required for code verification and long-term maintenance costs (which are also becoming a growing concern in the context of reproducibility and sustainable scientific computing). Hence, no method is always the best with respect to all the criteria. However, the choice of the optimum tool is still heavily dependent on the PDE under consideration, dimensionality, regularity of the solution, available computation resources, and, most importantly, the desired confidence and interpretability of the predictions made by the tool.

To make the foregoing comparison more complete, a summary of the different regimes, geometric treatment, merits and drawbacks of representative reduced-order, meshfree, operator-learning and hybrid PDE solution paradigms is given in Table 1.

Table 1 - Comparative synthesis of representative PDE solution methods

DOI: https://doi.org/10.60797/itech.2026.11.2.1

Paradigm	Core idea	Best-fit regime	Geometry / mesh handling	Key strengths	Main limitations and compute notes
Reduced Basis Methods (RBM) [13], [14], [15]	Projection-based reduced-order model built on a high-fidelity discretization.	Repeated queries for parametrized PDEs, optimal control, and inverse problems.	Medium; strongest when parametrization and the reference model are well defined.	Certified error estimates; offline/online split; strong classical numerical grounding.	Intrusive and less flexible for rapidly changing geometries; best when the solution manifold is smooth.
DGM / high-dimensional deep solvers [16], [17]	Train a neural ansatz on randomly sampled space-time points instead of a mesh.	High-dimensional PDEs where classical grids are infeasible, especially finance and control.	Meshfree, but typically demonstrated on simpler domains rather than irregular engineering geometries.	Avoids explicit meshes; demonstrated at very high dimension in representative studies.	Per-problem training cost is high; optimization replaces, rather than removes, heavy computation.
PINNs [18], [19], [20], [21], [22]	Neural solution representation constrained by PDE residuals, initial conditions, and boundary conditions.	Forward and inverse problems with sparse data or strong physics constraints.	Meshfree and flexible; can encode Dirichlet, Neumann, Robin, and periodic conditions.	Data-efficient and physically informed; natural for inverse and hybrid data-physics workflows.	Training pathologies include spectral bias, gradient imbalance, and causality issues; weak on high-frequency or multiscale solutions without advanced strategies.
DeepONet [22], [23], [24]	Operator learning with branch and trunk networks mapping input functions to output functions.	Many-query parametric PDE families where amortized inference matters.	Medium-high; can pair with CNN or GNN branches and is not tied to one geometry class.	Fast reuse after training; robust in noisy and complex settings.	Needs supervised training data and architecture tuning; upfront training cost can be substantial.
FNO [25], [26], [27], [28]	Neural operator with an integral kernel	Structured-grid parametric PDE families needing	Low-medium; strongest on regular domains	Very fast inference and zero-shot super-	Performance often degrades with noisy inputs

 © Authors of the article / Authors of the article

Paradigm	Core idea	Best-fit regime	Geometry / mesh handling	Key strengths	Main limitations and compute notes
	parameterized in Fourier space.	fast inference.	and grid-like data.	resolution in benchmark settings.	or complex geometries; less natural for irregular domains.
PINO [29], [30]	Hybrid neural operator combining data supervision with PDE constraints, often at higher resolution.	Operator learning when physical validity and high-resolution fidelity matter.	Medium; inherits operator-learning structure while adding physics constraints.	Better generalization and physical validity than pure FNO, while retaining large speedups.	Still depends on operator-learning infrastructure and fine-tuning; not as geometry-native as graph-mesh approaches.
MeshGraphNets [31], [32], [33]	Graph-network simulator operating directly on meshes and allowing adaptive remeshing.	Irregular geometries, adaptive meshes, mechanics, cloth, and fluid problems.	Very high; built for unstructured irregular meshes and adaptive discretization.	Resolution-independent dynamics; adaptivity concentrates compute where gradients are strong.	Requires graph-based training data and rollout design; lacks classical certification.
MP-PDE solvers [34], [35]	Autoregressive message-passing solver with stability-oriented training.	Generalization across resolution, topology, geometry, discretization regularity, and boundary conditions.	Very high; graph representation supports irregular sampling, geometry, and topology changes.	Fast, stable, and accurate in 1D and 2D tests; explicitly connected to FDM, FVM, and WENO ideas.	Training remains nontrivial and error bounds are not certified; best viewed as a neural-numerical hybrid.
PDEformer [36], [37]	Pretrained graph-transformer plus implicit neural representation for cross-family PDE solving.	Zero-shot or few-shot transfer and inverse coefficient recovery across one-dimensional PDE families.	Currently low in geometry scope because published evidence is still largely one-dimensional.	Promising foundation-model direction with zero-shot transfer and rapid fine-tuning.	Current evidence remains mostly one-dimensional, and pretraining is heavy relative to narrower expert models.

The solvers of the Neural PDE are supposed to be compared by the accuracy but also in regard to the hardware needs [32]. The vast majority of neural methods are primarily trained on the GPU. As the instance, DGM was trained on 6 GPU nodes, and training cost was high [40]. The neural operators were run on one NVIDIA V100 (16 GB) [43]. MP-PDE also required about 12-24 hours on an RTX 2080 Ti GPU [39]. Nevertheless, inference can be significantly quicker than classical solvers following training. In one of its benchmarks, FNO had 0.005 s inference versus 2.2 s with a conventional solver [44]. MeshGraphNets also accelerated well, particularly when there is a GPU in place [47]. Therefore, classical solvers are still easier to use in CPU-based workflows, and neural solvers are more appealing in many-query GPUs [45]. As such, the training hardware, inference hardware, runtime, and memory usage of each method should be reported in the review.

Representative application areas

Computational fluid dynamics, turbulence. The high-fidelity simulation of incompressible and compressible flows is a great source of innovation for new methods of numerically solving partial differential equations. Naturally, for a large number of applications in engineering and geophysics, finite-volume and finite-element discretization are by far the most popular workhorses, largely because of the good compromise of robustness, conservation properties and geometric flexibility offered by these discretization. Their performance is often improved through the use of adaptive mesh refinement (AMR), to focus the high-resolution in dynamically significant areas, and other computational iteratively challenging reconstruction schemes, such as multigrid acceleration, to decrease the computation time for their solvers, and turbulence-modelling schemes developed in variational multiscale frameworks, which attempt to formally divide and always consistently represent the flow with the resolved and the unresolved structures [12]. This combination of elements — both from the academic side and from the point of view of the real-life application — represents an old rule in CFD: predicting it correctly requires, in general, not only a good discretization, but also solver technology and modelling assumptions that guarantee that the solution remains stable under a wide range of Reynolds numbers and flow regimes. In recent years, AI-driven approaches have begun to influence the work

 © Authors of the article / Authors of the article

practices of CFD not just as a replacement of classical solvers, but rather as a specialised method for specific tasks. Specifically, machine learning models are nowadays gaining more and more importance as surrogates to subgrid scale closures, data-driven reduced order models for faster prediction of flows, components enhancing any aspect of the numerical pipeline (e.g., subgrid scale closures, parameter estimation) etc. Although these models can yield a significant amount of computational advantages, their accuracy, of course, relies on the scope of their training, and apart from considering physical limitations [6], [11] (these being an active field of investigation), their use in rolling out SDGs is still to come.

Solid Mechanics/Fracture. Most of the simulations used in industry that involve elasticity, plasticity and wave propagation in solids are based on the classical finite element method (FEM) including the high-order and discontinuous Galerkin (DG) methods. Meshfree and material point method (MPM) formulations are very useful in problems of large deformations, impact and fracture where remeshing would be prohibitive [9], [10] and in hybrid schemes, these are used together with traditional FEM in different sub-domains of the computational domain. In the case of material modelling, stress-strain responses are being investigated via emerging operator-learning and physics-informed neural network (PINN) based methods in order to evaluate them quickly [4], [41].

Fluid mechanics of Biomedical flows and transportation. Patient-specific vascular networks, or even the transport of pharmaceutical or signaling molecules, are typically modeled via finite-volume or finite-element solutions on a complex geometry obtained from medical imaging data [12]. To achieve clinically relevant time scales of such simulations, adaptive refinement, multigrid preconditioning and reduced-order models are of major importance. Data driven surrogates and PINNs have shown great potential to integrate sparse measurements and handle parameter uncertainty and real time or near real, time decision support [4], [11], [48].

Financial Partial Differential equations and Quantum problems. In the field of quantitative finance, partial differential equations formulations play a fundamental role, and in particular, the high dimensional Black Scholes-type models for multiple asset derivatives and similar stochastic partial differential equations models of risk factors. Numerically, these problems are challenging in that the computational cost rapidly grows as the number of dimensions is increased, an effect known as the "curse of dimensionality," and the simple possibility of mesh refinement for the solution of multisets problems is limited [1]. As a result, the focus of recent research has shifted more towards the emergence of neural network surrogates, physics-informed neural networks (PINNs) and operator learning architectures as alternatives or supplements to the traditional solvers. What has motivated them from an academic point of view is their potential to map high-dimensional solution manifold spaces into comparatively low-dimensional parameterizations, which can be evaluated quickly once trained, e.g., in pricing or calibration, but also in repeated risk queries with regard to varying market scenarios [4], [5], [6], [38], [42]. At the same time, critical methodological issues are the extent of the validity of such models when the distribution shifts, under rare events regimes or when market dynamics change — the practical applications of tail risks and extrapolation are also important. But at the same time, there are system level considerations which are equally important. The questions around scaling such complexes as the aforementioned load balancing in the adaptive and heterogeneous segment discoveries within the more challenging computational workflows or the resilience and fault tolerance of algorithms scaling at other extreme level or the energy efficiency of the networks or training and inference pipelines [44], [45], [46] are nontrivial.

Uncertainty quantification (UQ) is also a current bottleneck, and while probabilistic formulations can offer valuable information with regards to credibility, risk, etc, UQ on complex PDE models is often expensive and there is a strong incentive in the research community to work towards approaches that enable realisation of such models which involve integration of probabilistic modelling and efficient sampling, reduced order modelling, operator learning surrogates, etc, in a way that will control cost without compromising the statistical reliability of the results.

Conclusion

Modern numerical methods for the solution of partial differential equations are a vast and evolving arsenal, and there is a large body of theoretical and experimental evidence that supports the lack of a uniform paradigm that is universally found to be best for all classes of PDEs, geometries, and definition criteria. In this regard, classical finite-difference, finite-element, and finite-volume methods remain predictors of last resort for lack of maturity, efficiency, and well-justified analytical background [12].

Simultaneously, spectral, meshfree, adaptive and multiscale methods greatly extend the range of practicality of PDE simulation (e.g. in attaining superior accuracy for smooth fields, higher flexibility with respect to geometrical domains and efficient resolution of localised or multiscale phenomena which cannot generally be captured by traditional uniform discretisation [2], [3], [7], [9]).

More recently, solvers based on the Al — the most effective for large dimensions inverse and parametric problems — and operator learning methods have become influential developments, especially for high dimensional, inverse and parametric problems, for which one needs repetitive evaluation, or fast exploration of families of solutions. These methods can provide significant computational benefits and new modelling capabilities but usually do not yet provide the robustness, interpretability and theoretical guarantees supporting the routine use of mature discretization schemes in a safety-critical context [4], [5], [6], [11]. Academically, this gap suggests a key issue at the heart of the debate: between their impressive greater empirical performance and strict notions of stability, generalization, conservation, and error control, the deployment of such systems on a large scale cannot be justified.

 © Authors of the article / Authors of the article

Конфликт интересов

Не указан.

Рецензия

Все статьи проходят рецензирование. Но рецензент или автор статьи предпочли не публиковать рецензию к этой статье в открытом доступе. Рецензия может быть предоставлена компетентным органам по запросу.

Conflict of Interest

None declared.

Review

All articles are peer-reviewed. But the reviewer or the author of the article chose not to publish a review of this article in the public domain. The review can be provided to the competent authorities upon request.

Список литературы на английском языке / References in English

1. Evans L.C. Partial Differential Equations / L.C. Evans. — 2nd ed. — Providence : American Mathematical Society, 2010. — 662 p.

2. Canuto C. Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics / C. Canuto, M.Y. Hussaini, A. Quarteroni. — 2nd ed. — Berlin : Springer, 2014. — 564 p.

3. Fasshauer G.E. Meshfree Approximation Methods with MATLAB / G.E. Fasshauer. — Singapore : World Scientific, 2007. — 528 p.

4. Raissi M. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations / M. Raissi, P. Perdikaris, G.E. Karniadakis // Journal of Computational Physics. — 2019. — Vol. 378. — P. 686–707.

5. Lu L. DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators / L. Lu, P. Jin, G.E. Karniadakis // Nature Machine Intelligence. — 2021. — Vol. 3, № 3. — P. 218–229.

6. Li Z. Fourier neural operator for parametric partial differential equations / Z. Li, N. Kovachki, K. Azizzadenesheli [et al.] // International Conference on Learning Representations (ICLR). — 2021. — 25 p.

7. Berger M.J. Local adaptive mesh refinement for shock hydrodynamics / M.J. Berger, P. Colella // Journal of Computational Physics. — 1989. — Vol. 82, № 1. — P. 64–84.

8. Briggs W.L. A Multigrid Tutorial / W.L. Briggs, V.E. Henson, S.F. McCormick. — 2nd ed. — Philadelphia : SIAM, 2000. — 193 p.

9. Belytschko T. Meshless methods: An overview and recent developments / T. Belytschko, Y. Krongauz, D. Organ // Computer Methods in Applied Mechanics and Engineering. — 1996. — Vol. 139, № 1–4. — P. 3–47.

10. Sulsky D. Application of a particle-in-cell method to solid mechanics / D. Sulsky, S.-J. Zhou, H.L. Schreyer // Computer Physics Communications. — 1994. — Vol. 87, № 1–2. — P. 236–252.

11. Karniadakis G.E. Physics-informed machine learning / G.E. Karniadakis, I.G. Kevrekidis, L. Lu [et al.] // Nature Reviews Physics. — 2021. — Vol. 3, № 6. — P. 422–440.

12. Quarteroni A. Numerical Mathematics / A. Quarteroni, R. Sacco, F. Saleri. — 2nd ed. — Berlin : Springer, 2007. — 657 p.

13. Quarteroni A. Reduced Basis Methods for Partial Differential Equations: An Introduction / A. Quarteroni, A. Manzoni, F. Negri. — Cham : Springer, 2016.

14. Hesthaven J.S. Certified Reduced Basis Methods for Parametrized Partial Differential Equations and Applications / J.S. Hesthaven, G. Rozza, B. Stamm. — Cham : Springer, 2016.

15. Benner P. A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems / P. Benner, S. Gugercin, K. Willcox // SIAM Review. — 2015. — Vol. 57, № 4. — P. 483–531.

16. Sirignano J. DGM: A deep learning algorithm for solving partial differential equations / J. Sirignano, K. Spiliopoulos // Journal of Computational Physics. — 2018. — Vol. 375. — P. 1339–1364.

17. Han J. Deep Learning-Based Numerical Methods for High-Dimensional Parabolic Partial Differential Equations and Backward Stochastic Differential Equations / J. Han, A. Jentzen // Communications in Mathematics and Statistics. — 2017. — Vol. 5, № 4. — P. 349–380.

18. Cuomo S. Scientific Machine Learning Through Physics-Informed Neural Networks: Where We Are and What's Next / S. Cuomo, V. Schiano Di Cola, F. Giampaolo [et al.] // Journal of Scientific Computing. — 2022. — Vol. 92. — Art. 88.

19. Wang S. When and why PINNs fail to train: A neural tangent kernel perspective / S. Wang, X. Yu, P. Perdikaris // Journal of Computational Physics. — 2022. — Vol. 449. — Art. 110768.

20. Cai S. Physics-informed neural networks (PINNs) for fluid mechanics: A review / S. Cai, Z. Mao, Z. Wang [et al.] // Acta Mechanica Sinica. — 2021. — Vol. 37, № 12. — P. 1727–1738.

21. Lu L. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators / L. Lu, P. Jin, G. Pang [et al.] // Nature Machine Intelligence. — 2021. — Vol. 3, № 3. — P. 218–229.

22. Kovachki N.B. Operator learning: Algorithms and analysis / N.B. Kovachki, S. Lanthaler, A.M. Stuart // Handbook of Numerical Analysis. — 2024. — Vol. 25. — P. 419–467.

23. Kovachki N. Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs / N. Kovachki, Z. Li, B. Liu [et al.] // Journal of Machine Learning Research. — 2023. — Vol. 24, № 89. — P. 1–97.

24. Li Z. Fourier Neural Operator for Parametric Partial Differential Equations / Z. Li, N. Kovachki, K. Azizzadenesheli [et al.] // International Conference on Learning Representations (ICLR). — 2021.

25. Kovachki N. Neural Operator: Learning Maps Between Function Spaces / N. Kovachki, Z. Li, B. Liu [et al.]. — 2021. — arXiv:2108.08481.

26. Tran A. Factorized Fourier Neural Operators / A. Tran, A. Mathews, L. Xie // International Conference on Learning Representations (ICLR). — 2023.

 © Authors of the article / Authors of the article

27. Li Z. Geometry-Informed Neural Operator for Large-Scale 3D PDEs / Z. Li, N. Kovachki, C. Choy [et al.] // Advances in Neural Information Processing Systems. — 2023. — Vol. 36.

28. Li Z. Physics-Informed Neural Operator for Learning Partial Differential Equations / Z. Li, H. Zheng, N. Kovachki [et al.]. — 2021. — arXiv:2111.03794.

29. Calvello E. Continuum Attention for Neural Operators / E. Calvello, N.B. Kovachki, M.E. Levine // Journal of Machine Learning Research. — 2025. — Vol. 26. — P. 1–52.

30. Pfaff T. Learning Mesh-Based Simulation with Graph Networks / T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez [et al.] // International Conference on Learning Representations (ICLR). — 2021.

31. Pfaff T. Learning Mesh-Based Simulation with Graph Networks / T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez. — 2021. — URL: https://openreview.net/forum?id=roNqYL0_XP (accessed: 06.04.2026).

32. Sanchez-Gonzalez A. Learning to Simulate Complex Physics with Graph Networks / A. Sanchez-Gonzalez, J. Godwin, T. Pfaff [et al.] // Proceedings of the 37th International Conference on Machine Learning. — 2020. — Vol. 119. — P. 8459–8468.

33. Brandstetter J. Message Passing Neural PDE Solvers / J. Brandstetter, D.E. Worrall, M. Welling // International Conference on Learning Representations (ICLR). — 2022.

34. Lienen M. Learning the Dynamics of Physical Systems from Sparse Observations with Finite Element Networks / M. Lienen, S. Gunnemann // International Conference on Learning Representations (ICLR). — 2022.

35. Ye Z. PDEformer: Towards a Foundation Model for One-Dimensional Partial Differential Equations / Z. Ye, X. Huang, L. Chen [et al.]. — 2024. — arXiv:2402.12652. — URL: https://arxiv.org/abs/2402.12652 (accessed: 06.04.2026).

36. Wu H. Transolver: A Fast Transformer Solver for PDEs on General Geometries / H. Wu, H. Luo, H. Wang [et al.] // Proceedings of the 41st International Conference on Machine Learning. — 2024. — Vol. 235. — P. 53681–53705.

37. Koumoutsakos P. Machine learning and partial differential equations: benchmark, simplify, and discover / P. Koumoutsakos // Data-Centric Engineering. — 2025. — Vol. 6. — P. e29.

38. Khoo Y. Solving parametric PDE problems with artificial neural networks / Y. Khoo, J. Lu, L. Ying // European Journal of Applied Mathematics. — 2021. — Vol. 32, № 3. — P. 421–435.

39. Han J. Solving high-dimensional partial differential equations using deep learning / J. Han, A. Jentzen, W. E // Proceedings of the National Academy of Sciences of the United States of America. — 2018. — Vol. 115, № 34. — P. 8505–8510.

40. Ghattas O. Learning physics-based models from data: Perspectives from inverse problems and model reduction / O. Ghattas, K. Willcox // Acta Numerica. — 2021. — Vol. 30. — P. 445–554.

41. Raissi M. Deep Hidden Physics Models: Deep Learning of Nonlinear Partial Differential Equations / M. Raissi // Journal of Machine Learning Research. — 2018. — Vol. 19, № 25. — P. 1–24.

42. Long Z. PDE-Net: Learning PDEs from Data / Z. Long, Y. Lu, X. Ma // Proceedings of the 35th International Conference on Machine Learning. — 2018. — Vol. 80. — P. 3208–3216.

43. Ferziger J.H. Computational Methods for Fluid Dynamics / J.H. Ferziger, M. Peric, R.L. Street. — 4th ed. — Cham : Springer, 2020. — 596 p.

44. LeVeque R.J. Finite Volume Methods for Hyperbolic Problems / R.J. LeVeque. — Cambridge : Cambridge University Press, 2002. — 580 p.

45. Donea J. Finite Element Methods for Flow Problems / J. Donea, A. Huerta. — Chichester : John Wiley & Sons, 2003. — 362 p.

46. Hughes T.J.R. Large Eddy Simulation and the variational multiscale method / T.J.R. Hughes, L. Mazzei, K.E. Jansen // Computing and Visualization in Science. — 2000. — Vol. 3. — P. 47–59.

47. Brunton S.L. Machine Learning for Fluid Mechanics / S.L. Brunton, B.R. Noack, P. Koumoutsakos // Annual Review of Fluid Mechanics. — 2020. — Vol. 52. — P. 477–508.

48. Duraisamy K. Turbulence Modeling in the Age of Data / K. Duraisamy, G. Iaccarino, H. Xiao // Annual Review of Fluid Mechanics. — 2019. — Vol. 51. — P. 357–377.