Hybrid AI-Quantum Computational Framework for Efficient Eddy Current Repulsion Force Simulation

Zhou Zhang[1,*], Yeong Ryu[1], Wenhai Li[1] and Jiayin Wang[2]

[1] Department of Mechanical Engineering Technology, SUNY Farmingdale State College, Farmingdale, NY 11735, USA

[2] School of Computing, College of Science and Mathematics, Montclair State University, Montclair, NJ 07043, USA

INFORMATION

Keywords:
Eddy current simulation
quantum-assisted finite element method
physics-informed neural networks (PINNs)
hybrid AI-quantum computing

DOI: 10.23967/j.rimni.2026.10.74579

SCIPEDIA Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

Revista Internacional Métodos numéricos para cálculo y diseño en ingeniería RIMNI

In cooperation with CIMNE

Hybrid AI-Quantum Computational Framework for Efficient Eddy Current Repulsion Force Simulation

Zhou Zhang[1,*], **Yeong Ryu**[1], **Wenhai Li**[1] **and Jiayin Wang**[2]

[1]Department of Mechanical Engineering Technology, SUNY Farmingdale State College, Farmingdale, NY 11735, USA

[2]School of Computing, College of Science and Mathematics, Montclair State University, Montclair, NJ 07043, USA

ABSTRACT

Eddy current repulsion forces are essential to the operation of magnetic levitation, electromagnetic braking, and wireless power transfer systems, yet their accurate simulation remains computationally intensive due to the complexity of Maxwell's equations and the need for fine-resolution meshing in transient, high-frequency domains. This paper presents a hybrid AI-Quantum computational framework that accelerates eddy current simulation by combining adaptive meshing via Physics-Informed Neural Networks for adaptive meshing, quantum-assisted solvers (Harrow-Hassidim-Lloyd algorithm and Variational Quantum Eigensolver) for selected finite element subdomains, and GPU-accelerated finite element methods, and reduced-order modeling based on Proper Orthogonal Decomposition and Quantum Autoencoders. The framework selectively routes well-conditioned sparse subblocks to quantum solvers while retaining classical GPU-based methods elsewhere. Validation across three representative applications—electromagnetic braking, magnetic levitation, and wireless power transfer—demonstrates that the hybrid solver maintains accuracy within 5%–8% of full FEM results while reducing computational cost by 1.3×–2× speedup in the tested scenarios. These results confirm the feasibility of integrating AI and near-term quantum computing into electromagnetic simulation workflows and provide guidance on complexity, resource requirements, and scalability.

OPEN ACCESS

Received: 14/10/2025

Accepted: 22/12/2025

Published: 29/05/2026

DOI
10.23967/j.rimni.2026.10.74579

Keywords:
Eddy current simulation
quantum-assisted finite element method
physics-informed neural networks (PINNs)
hybrid AI-quantum computing

1 Introduction

Eddy current repulsion force is a fundamental electromagnetic phenomenon arising when a time-varying magnetic field induces circulating currents—eddy currents, in a conductive material [1]. These eddy currents generate their own magnetic fields, which, according to Lenz's law, oppose the change in the original magnetic field. The resulting force is a repulsive effect that has been harnessed in a wide range of engineering systems.

The crucial applications include (refer to Fig. 1): (1) Magnetic Levitation (MagLev) [2,3]. High-speed transportation and contactless bearing systems achieve frictionless motion and stable levitation. (2) Electromagnetic Braking [4,5]. Used in trains, amusement rides, and high-performance vehicles,

*Correspondence: Zhou Zhang (zhangz@farmingdale.edu)

where contactless braking ensures smoother deceleration and reduced mechanical wear. (3) Wireless Power Transfer (WPT) [6,7]: Efficient and safe transfer of power in electric vehicle charging and biomedical implants depends on managing and minimizing parasitic eddy current losses.

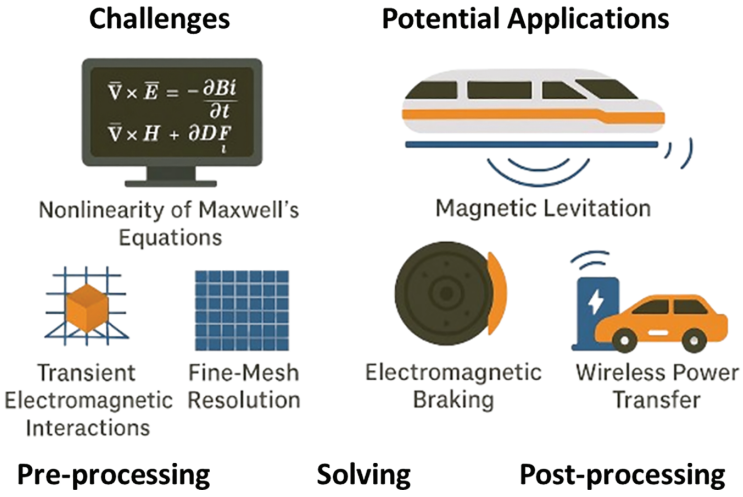

Figure 1: Challenges and potential applications

As shown in Fig. 1, there are critical challenges in the eddy current repulsion forces simulation: (1) Nonlinearity of Maxwell's Equations [8]: Particularly under transient conditions (both magnetic and electric fields evolve rapidly and interact with each other). (2) Skin and Proximity Effects [9]: High-frequency excitations lead to steep field gradients near conductive surfaces, requiring extremely fine mesh resolution to resolve accurately. (3) Large-Scale 3D Domains [10]: Realistic simulations often involve complex geometries with millions of degrees of freedom (DOF), which increases computational demand dramatically. (4) Coupled Multiphysics Interactions [11,12]: In many systems, electromagnetic behavior is coupled with mechanical or thermal effects, further complicating numerical simulation.

Traditional simulation workflows typically rely on FEM which offers a robust framework for solving Maxwell's equations in arbitrary geometries [13–15]. It is computationally intensive and struggles to balance between fine resolution and acceptable runtime. So, it becomes critical when fast iteration, real-time control, or large design space exploration is required. Therefore, to overcome these challenges, we propose a hybrid AI-Quantum-GPU computational framework that innovatively integrates the following key technologies: (1) Physics-Informed Neural Networks (PINNs) [16]: Serve as an intelligent meshing controller, dynamically generating adaptive mesh density fields to minimize error and reduce computation. (2) Quantum-Assisted Solvers: Including the Harrow-Hassidim-Lloyd (HHL) algorithm [17] and Variational Quantum Eigensolver (VQE) [18,19], these solvers accelerate matrix inversion and eigenvalue computations in the FEM solution process. (3) Reduced-Order Modeling (ROM) with Quantum Autoencoders (QAE) [20]: Compresses full-scale solutions into a low-dimensional latent space for real-time force and field prediction with fidelity. (4) GPU-Accelerated FEM Solvers with Dynamic Load Balancing [21,22]: Achieves scalable, high-performance matrix assembly and solution across large and heterogeneous simulation domains.

This framework is designed as a practical and extensible system for accelerating real-world eddy current simulation, offering enhanced efficiency, improved accuracy, and real-time capability. We discuss its architecture, implementation, and validation through representative case studies, demonstrating significant improvements over traditional methods in both speed and accuracy.

2 Related Work

Numerical simulation of eddy current phenomena is essential such as magnetic levitation, braking systems, and wireless power transfer [23,24]. While advances in FEM, meshing, and hardware acceleration have improved accuracy, existing methods still face challenges with high-frequency, nonlinear, and real-time scenarios [25–27]. This section reviews key developments in FEM, adaptive meshing, quantum solvers, AI-based ROM, and GPU computing, motivating the need for our hybrid framework.

2.1 FEM-Based Eddy Current Simulation

FEM remains the standard for simulating eddy current phenomena due to its accuracy and flexibility in handling complex geometries, material heterogeneity, and boundary conditions [28,29]. Using quasi-static Maxwell's equations—typically through magnetic vector potential or curl-curl formulations—FEM discretizes the problem into large, sparse matrix systems. Tools like ANSYS Maxwell, COMSOL, and Elmer FEM offer robust implementations but can become computationally intensive, especially in high-resolution or frequency-dependent scenarios [30,31].

To alleviate these burdens, numerous improvements have been proposed in the literature and in practice: (1) Domain Decomposition Methods (DDM) [32–34]: These methods divide the simulation domain into smaller subregions, which can be solved in parallel, reducing memory consumption and improving scalability. However, efficient domain interfaces and convergence control remain nontrivial. (2) Multigrid and Algebraic Multigrid (AMG) Solvers [35–37]: These preconditioners accelerate convergence of iterative solvers by solving coarser approximations of the system. Though powerful, they can introduce numerical instability in highly anisotropic or heterogeneous media without careful tuning. (3) Model-Order Reduction (MOR): Techniques such as Proper Orthogonal Decomposition (POD) [38,39], Krylov subspace projection, and reduced basis methods are employed to approximate full-order models with lower-dimensional surrogates. While effective in many cases, they are often limited to linear or mildly nonlinear systems, and their offline construction remains costly. (4) High-Order and Adaptive Elements [40,41]: Using higher-order basis functions (e.g., quadratic, cubic) can improve accuracy without increasing mesh density. Adaptive meshing techniques that refine based on field gradients or residual errors help reduce unnecessary computation in uniform regions.

Despite advances, traditional FEM solvers remain limited in real-time applications, especially for complex 3D, high-frequency, or multi-physics problems. Their long runtimes and reliance on computationally expensive remeshing cycles hinder use in feedback control or rapid design tasks. Recent efforts to enhance FEM with AI-driven meshing and HPC acceleration show promise but remain fragmented. This work proposes a unified, scalable hybrid framework that integrates FEM with artificial intelligence, quantum solvers, and GPU acceleration to address these persistent challenges.

2.2 Adaptive Meshing and Error-Driven Refinement

Adaptive meshing enhances accuracy without the cost of uniformly fine grids. Traditional methods use a posteriori error estimators—like residual-based indicators [42] or Zienkiewicz-Zhu

techniques [43]—to guide refinement after an initial FEM solution. However, this iterative process is computationally expensive, involving repeated mesh updates and matrix reassembly. Recent approaches embed physics knowledge into refinement but still rely on rule-based strategies. Emerging techniques now leverage PINNs, originally designed to solve PDEs, to predict mesh density functions instead. By identifying regions of high error or gradient in advance, PINNs enable efficient, data-driven mesh generation—reducing the need for costly iterative remeshing and enhancing scalability.

2.3 Quantum Computing in Numerical Simulation

Quantum computing offers significant theoretical advantages for solving linear systems and eigenvalue problems. The HHL algorithm (2009) [16] showed that, under ideal conditions—such as sparsity, Hermitian matrices, and low condition numbers—quantum solvers can outperform classical methods exponentially. Later studies applied these algorithms to electromagnetic and fluid problems, primarily as proofs of concept using simulators. More practical hybrid algorithms like the VQE leverage classical optimizers to tune quantum circuits, making them suitable for frequency-domain modal analysis. Despite hardware limitations in current NISQ devices, quantum solvers can still contribute effectively when used in hybrid workflows—solving small, well-conditioned subdomains while classical methods handle the rest. Our work follows this philosophy, integrating quantum solvers into blockwise FEM operations such as local matrix inversion and modal decomposition, yielding promising results in simulation.

2.4 AI-Based Reduced-Order Modeling

Model reduction is essential for accelerating simulations in real-time control and design optimization. Classical techniques like POD, balanced truncation, and reduced basis methods simplify high-dimensional problems by projecting them onto low-dimensional subspaces. However, these methods often fall short in handling strong nonlinearities or multiparameter variations, and they typically require expensive full-order snapshots. AI-driven ROM approaches [44] address these limitations by using neural networks—such as autoencoders and convolutional models [45]—to learn compact representations that better capture nonlinear dynamics. Recently, QAE have emerged as a promising tool to compress and reconstruct high-dimensional solution data using quantum circuits, enabling efficient simulation in quantum latent spaces. Our framework integrates both POD and QAE to deliver fast and accurate predictions of field distributions and eddy current forces.

2.5 GPU-Accelerated FEM and Parallel Solvers

High-performance computing for FEM has increasingly leveraged GPU acceleration to expedite matrix operations such as element assembly, sparse matrix-vector multiplication, and iterative solvers. Frameworks like cuSPARSE [46], cuBLAS [47], and CUDA-enabled PETSc [48] have demonstrated significant speedups in large-scale simulations. However, maintaining performance in scenarios with adaptive meshing or dynamic fields requires careful workload balancing. To address this, recent approaches use graph partitioning and asynchronous data handling. Our framework incorporates dynamic GPU load balancing alongside PINN-driven adaptive meshing to sustain high throughput under evolving simulation conditions.

2.6 Synthesis and Research Gap

While prior studies have advanced AI acceleration, quantum solvers, and GPU-based FEM individually, few have combined them into a unified, practical framework. This paper addresses that gap

by introducing and validating a hybrid system that integrates AI-driven mesh adaptation, quantum-assisted solvers for sparse and modal problems, GPU-parallel FEM for efficient computation, and reduced-order modeling for real-time scalability. This cohesive approach enables high-fidelity, real-time eddy current simulations across applications such as electromechanical systems, sustainable energy, and smart robotics.

Unlike prior studies that focus on a single acceleration mechanism—such as AI-based surrogates, GPU-only FEM, or proof-of-concept quantum solvers—our work integrates four components into a unified pipeline: (i) PINN-driven adaptive meshing, (ii) blockwise quantum solvers for sparse subdomains, (iii) GPU-accelerated FEM with dynamic load balancing, and (iv) ROM based on POD and QAE. To the best of our knowledge, this is the first framework that combines AI, quantum, and GPU technologies in a coherent architecture specifically tailored for eddy current repulsion problems. Table 1 summarizes the qualitative differences between representative prior approaches and the proposed framework in terms of meshing, solver composition, and hardware utilization.

Table 1: Qualitative comparison between representative existing approaches and the proposed hybrid AI-Quantum-GPU framework

Method category	Adaptive meshing	Quantum subdomain solvers	GPU acceleration	ROM with QAE
Classical FEM (CPU/GPU Only)	No (static mesh)	No	Optional (GPU FEM)	No
AI-FEM Coupled Methods (PINN or ML-Based)	Yes (AI-driven meshing or surrogates)	No	Optional	Sometimes (classical autoencoders)
Hybrid Quantum-Classical FEM (HHL/VQE Papers)	No	Yes (global or partial matrices)	No (typically)	No
Reduced-Order Models (POD/AE-Based)	No	No	No	Yes (classical AE)
Proposed Hybrid AI-Quantum-GPU Framework	Yes (PINN-based)	Yes (block-level HHL/VQE)	Yes (multi-GPU with load balancing)	Yes (POD + QAE)

3 Proposed Hybrid Computational Framework

To address the computational challenges of simulating eddy current repulsion force, we propose a hybrid framework that integrates four core components: (1) AI-enhanced adaptive meshing based on modified PINNs, (2) Quantum-assisted solvers for sparse linear systems and eigenvalue problems, (3) AI-accelerated ROM for dynamic simulation compression, and (4) GPU-parallelized FEM with dynamic load balancing.

Fig. 2, presents the complete data-flow of the proposed hybrid AI-Quantum-GPU framework. The workflow begins with the modified PINN, which employs physics residuals, geometric curvature, and PINN-based error-indicator losses to predict a spatially varying mesh-density field. This field is converted into an element-sizing function that guides a PINN-driven adaptive meshing stage, producing a graded triangular discretization that concentrates resolution in regions of strong electromagnetic variation. The mesh is then passed to the GPU-accelerated FEM pipeline, where element matrices are assembled and the global sparse system is constructed.

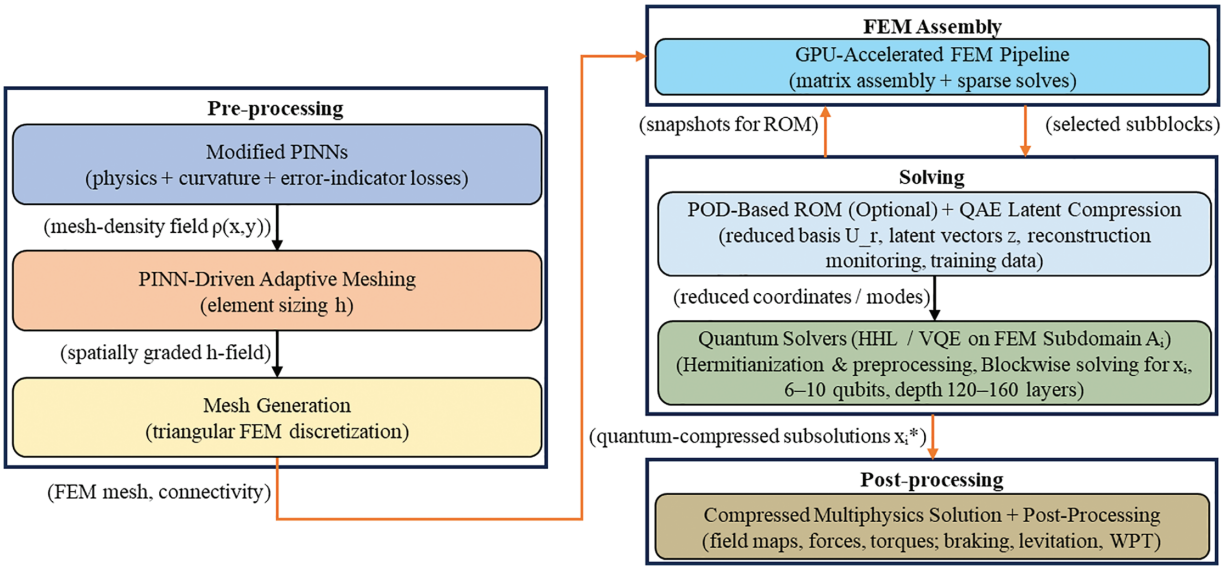

Figure 2: Hybrid AI–quantum–GPU computational framework

From this FEM stage, two key data streams are generated: (i) high-fidelity solution snapshots used for POD basis construction and QAE latent-space compression, and (ii) diagonal block submatrices obtained via domain decomposition. Each submatrix is evaluated for sparsity, size, and conditioning to determine whether it is suitable for quantum processing. Quantum-eligible blocks are solved using HHL or VQE on Qiskit, while the remaining blocks and the Schur complement system are solved using GPU-based iterative solvers. The ROM component uses the POD basis and QAE encodings to compress and reconstruct solution states, providing efficient inference for repeated or parameterized simulations. The quantum-refined, GPU-FEM, and ROM outputs are merged to form the complete multiphysics solution, which is post-processed to compute field distributions, repulsion forces, torques, and power-loss metrics for braking, levitation, and WPT applications.

Fig. 3 provides a step-by-step outline of the integrated workflow. It summarizes the main operations of the pipeline—PINN-guided meshing, GPU-FEM assembly, hybrid classical–quantum solving, and optional ROM compression—offering a high-level view of the computational sequence from model input to final solution.

```
# ======================================================
# Step 0: Input Data
# ======================================================
geometry      = load_geometry('model.stl')
materials     = load_material_properties('materials.json')
excitation    = load_excitation_conditions('excitation.json')
coarse_points = sample_domain(geometry)

# ======================================================
# Step 1: AI-Driven Adaptive Meshing (Modified PINN)
# ======================================================
# Predict continuous mesh-density field h(r)
density_field = PINN.predict_density(coarse_points)

# Convert density field → spatially varying element sizes → adaptive mesh
mesh = adaptive_mesher.generate(geometry, density_field)

# ======================================================
# Step 2: GPU-Based FEM Assembly
# ======================================================
fem = FEMModel(mesh, materials, excitation)
A, b = fem.assemble_system()        # GPU-parallelized element assembly
subblocks = domain_decompose(A)     # For hybrid classical-quantum solving

# ======================================================
# Step 3: Hybrid Classical-Quantum Solving
# ======================================================
solution_blocks = []

for A_qq, b_qq in subblocks:

    # Determine quantum eligibility from block size and condition number
    if is_quantum_eligible(A_qq):
        A_q = hermitianize(A_qq)
        x_q = run_quantum_solver(A_q, b_qq)  # HHL or VQE
        solution_blocks.append(x_q)

    else:
        x_c = gpu_iterative_solver(A_qq, b_qq)  # CG / GMRES on GPU
        solution_blocks.append(x_c)

# Merge block-level solutions into global vector
solution = merge_subdomain_solutions(solution_blocks, A, b)

# ======================================================
# Step 4: Optional Reduced-Order Modeling (ROM)
# ======================================================
if use_ROM:

    # Project full-order solution onto POD basis
    POD_basis = load_POD_basis('pod_modes.npy')
    modal_coeffs = project_to_POD(solution, POD_basis)

    # Compress + reconstruct using trained QAE
    latent_vec = QAE.encode(modal_coeffs)
    modal_reconstructed = QAE.decode(latent_vec)

    # Reconstruct full field
    solution = reconstruct_from_POD(modal_reconstructed, POD_basis)

# ======================================================
# Step 5: Post-Processing
# ======================================================
fields = compute_electromagnetic_fields(solution, mesh)
forces = compute_eddy_current_forces(fields)
torque = compute_braking_or_levitation_torque(fields)


power_loss = compute_WPT_losses(fields)

visualize_results(fields, forces, torque)

save_results({
    "solution": solution,
    "fields": fields,
    "forces": forces,
    "torque": torque,
    "power_loss": power_loss
})

validate_solution(fields, reference_data='benchmark.json')
```

Figure 3: Algorithm: hybrid AI-quantum FEM simulation pipeline for eddy current repulsion force modeling

The description of this Algorithm 1 as follows:

Algorithm 1: Integrated PINN–Quantum–GPU–ROM workflow for efficient eddy current simulation

Input: Geometry, materials, excitation, initial coarse mesh.

1. Train the modified PINN to predict the mesh-density field $h(r)$ (Section 3.1).
2. Generate the adaptive mesh from $h(r)$ and assemble FEM matrices on GPU (Section 3.4).
3. Apply domain decomposition to obtain block partition $\{A_{ij}\}$.
4. For each block A_{qq}:
   Estimate sparsity and condition number $\kappa(A_{qq})$.
   If $\kappa\left(A_{qq}\right) \leq \kappa_{th}$ and $size \leq N_{max}$: mark block as **quantum-eligible.**
   Else: route to GPU solution.
5. Solve quantum-eligible blocks using HHL or VQE on Qiskit (Section 3.2).
6. Solve all remaining blocks and the Schur complement using GPU-based iterative solvers.
7. Optionally project solutions onto the POD basis and compress using a QAE for ROM inference (Section 3.3).
8. Post-process electromagnetic fields and compute repulsion force, torque, and power-loss quantities.

Output: Approximate electromagnetic fields, repulsion forces, and derived physical quantities; optional ROM for rapid evaluation.

3.1 AI-Driven Adaptive Meshing with Modified PINNs

3.1.1 Theoretical Foundations

Adaptive meshing is critical for eddy current simulations due to the skin effect and steep magnetic field gradients. Traditional methods based on a posteriori error estimation iteratively refine the mesh after solving, but they are computationally intensive and unsuitable for real-time or large-scale applications.

PINNs introduced by Raissi et al. [49], offer a framework for incorporating physical laws directly into the training of neural networks. Instead of relying on large labeled datasets, PINNs use the governing PDEs (in this case, Maxwell's equations) as constraints in the network's loss function. This allows the network to learn representations that respect underlying physics [50–52]. In our framework, we do not use PINNs to solve the electromagnetic PDEs directly. Instead, we use a modified PINN as a mesh density predictor that learns a continuous scalar field $h(r)$ with (r: Global reduced order dimension), representing the desired element size at each point in the domain Ω, based on the local physical field properties.

3.1.2 Mesh-Density Loss Formulation

Define a mesh density function $h(r) \in (h_{min}, h_{max})$, Small values indicate finer mesh. PINN surrogate learns this function by minimizing a multi-objective loss function over Ω. We introduce a composite loss with three components:

$$\mathcal{L}_{total} = \alpha_1 \mathcal{L}_{physics} + \alpha_2 \mathcal{L}_{indicator} + \alpha_3 \mathcal{L}_{curvature}$$

$\mathcal{L}_{physics}$: Physics loss. Residual of Maxwell's equations to guide the network to assign finer mesh where the PDE residual is large:

$\mathcal{L}_{physics} = \frac{1}{N}\sum_{i=1}^{N}\left\|\nabla\times\left(\frac{1}{\mu}\nabla\times A_\theta(r_i)\right)+j\omega\sigma A_\theta(r_i)-J_s(r_i)\right\|^2$ with (r_i: Local reduced order dimension (block-wise).

$\mathcal{L}_{indicator}$: Error indicator loss. Based on the Zienkiewicz-Zhu error estimator or residual-based quantities from a coarse FEM solution:

$$\mathcal{L}_{indicator} = \frac{1}{N}\sum_{i=1}^{N}(\eta(r_i)-\hat{\eta}_\theta(r_i))^2$$

where η is the the estimated error and $\hat{\eta}_\theta$ is the network's prediction.

$\mathcal{L}_{curvature}$: Curvature loss. Penalizes under-resolving high-curvature regions in the geometry or fields:

$$\mathcal{L}_{curvature} = \frac{1}{N}\sum_{i=1}^{N}(\kappa(r_i)-\hat{\kappa}_\theta(r_i))^2$$

where κ is computed from the Hessian of the field or geometry.

This is capturing (i) physics residuals from the governing Maxwell formulation, (ii) error-indicator targets approximating FEM residual estimates on a coarse mesh, and (iii) curvature-based refinement promoting resolution near geometric or field gradients. This design steers mesh refinement directly where the underlying physics and derivative structure demand it.

3.1.3 PINN Architecture and Training Strategy

Inputs: Spatial coordinates $r = (x, y, z)$.

Outputs: Mesh density value $h(r)$.

Architecture: A fully connected feedforward neural network with sine activation (to capture high-frequency variations), residual connections, and gradient flow preservation.

Training: Losses are computed on a coarse grid or via surrogate physics. Curriculum learning guides the network—initially focusing on geometry curvature, then refining based on physics. Loss weights (α_1, α_2, α_3) are dynamically adjusted using convergence and local field variation metrics.

To visualize the multi-stage convergence behavior of the PINN training process, Fig. 4 presents the evolution of the curvature, physics-residual, and error-indicator losses over the course of training. The multi-stage training schedule (geometry curvature → physics residual → error indicator) results in stable convergence and effective mesh-density prediction. Loss weights α_i are dynamically adjusted based on convergence and spatial variation.

3.1.4 Mesh Generation and FEM Coupling

Once the network is trained, the predicted $h(r)$ is mapped to a spatial mesh using octree refinement or Delaunay-based triangulation, depending on the domain type (structured vs. unstructured). This mesh is then directly passed to the GPU-FEM solver. Optionally, the PINN is retrained or fine-tuned after one FEM pass using residual feedback, enabling a closed-loop adaptive process.

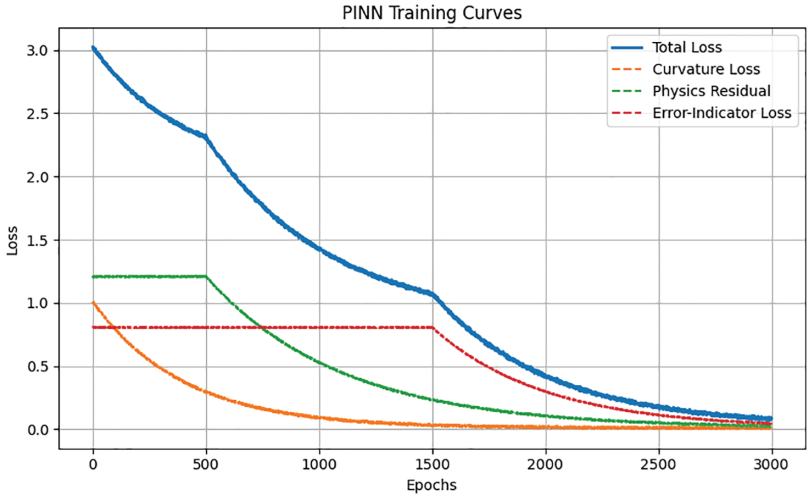

Figure 4: PINN training curves showing total loss, curvature loss, physics residual loss, and error-indicator loss across 3000 epochs

3.1.5 Summary of Modifications and Contributions

Key Contributions (refer to Table 2) that summarize how our approach extends conventional PINNs by introducing a physics–curvature–error composite loss, dynamic weighting, and closed-loop integration:

- Novel use of PINNs to drive adaptive meshing rather than solving PDEs directly.
- Mesh optimization integrates physics, error indicators, and curvature into a unified loss.
- Closed-loop feedback mechanism between FEM results and mesh predictor refinement.

Table 2: Modifications and contributions about AI-driven adaptive meshing with modified PINNs

Component	Traditional PINN	Modified PINN (Ours)
Purpose	Solves PDE	Predicts mesh density field
Loss function	PDE residuals only	PDE + curvature + error indicator
Architecture	Basic MLP	Residual blocks + sine activations
Adaptivity	Fixed loss terms	Dynamic weighting and curriculum learning
Coupling	Weak or standalone	Integrated with FEM and closed-loop refinement

3.2 Quantum-Assisted Matrix Solvers

3.2.1 Theoretical Foundations

In FEM-based eddy current simulations, discretizing Maxwell's equations yields a large, sparse, complex-valued linear system $Ax = b$. Where $A \in \mathbb{C}^{N\times N}$, is typically sparse and Hermitian (in frequency-domain formulations), x is the solution vector (e.g., vector potentials or nodal values). b is the excitation vector from current sources.

For large-scale 3D problems, solving this system becomes the dominant computational bottleneck [53], particularly during repeated solves, high-accuracy requirements, or fine mesh resolutions with millions of degrees of freedom. Quantum computing offers potential speedups for such sparse systems. This framework leverages two key quantum algorithms: HHL for linear system solutions and VQE for eigenvalue problems in modal and transient analyses.

3.2.2 HHL Algorithm for Linear System Solving

The HHL algorithm [54] is designed to solve a linear system $Ax = b$ by encoding vectors and operators into quantum states and using quantum phase estimation (QPE) to extract the solution. he HHL algorithm outputs a quantum state $|x\rangle$ that encodes the normalized solution x, i.e., $|x\rangle \propto A^{-1}|b\rangle$. This is achieved through four key quantum operations:

Input encoding. The vector b is first encoded into a normalized quantum state $|b\rangle$ using amplitude encoding:

$$|b\rangle = \frac{1}{\|b\|}\sum_{i=0}^{N-1} b_i |i\rangle$$

Efficient preparation of $|b\rangle$ is a necessary assumption for HHL. While exact amplitude encoding may require a quantum oracle or classical pre-processing, in our framework, we assume structured excitation vectors (e.g., from current sources with known distributions) that allow efficient encoding in simulation environments.

1. Quantum phase estimation [55]. It is applied to perform spectral decomposition of A. Since A is Hermitian, it can be decomposed into eigenpairs:

   $$A|u_j\rangle = \lambda_j |u_j\rangle$$

   The QPE operation encodes each eigenvalue into an ancilla register entangled with the corresponding eigenvector $|u_j\rangle$, yielding:

   $$|b\rangle = \sum_j \beta_j |u_j\rangle \rightarrow \sum_j \beta_j |u_j\rangle |\lambda_j$$

   This step assumes that a unitary operation e^{iAt} can be implemented efficiently, which is feasible for sparse FEM matrices via Hamiltonian simulation techniques.
2. Controlled rotation enables matrix inversion by rotating an ancilla qubit with an angle inversely related to eigenvalue λ_j.

   $$\sum_j \beta_j |u_j\rangle |\lambda_j \rightarrow \sum_j \frac{\beta_j}{\lambda_j} |u_j\rangle |\lambda_j\rangle |1 + \ldots$$

   The ancilla qubit scales each λ_j^{-1}, with thresholding or regularization used to discard near-zero eigenvalues and ensure stability.
3. Inverse QPE and measurement. Inverse QPE is applied to disentangle the eigenvalue register from the solution state, yielding the final quantum solution:

   $$|x\rangle = \sum_j \frac{\beta_j}{\lambda_j} |u_j\rangle$$

which encodes the normalized solution $x = A^{-1}b$. Observables (e.g., inner products, energy integrals, force components) can be extracted via quantum measurements, though full state tomography is impractical for large N.

The condition number and algorithmic complexity. The algorithm's efficiency is heavily influenced by the condition number $\kappa(A)$ that is defined as: $\kappa(A) = \|A\| \cdot \left\|A^{-1}\right\| = \frac{\lambda_{max}}{\lambda_{min}}$ (for Hermitian, positive-definite A). The total complexity of HHL scales as $O(\kappa^2 logN)$. Therefore, systems with large κ (e.g., due to sharp material contrasts or poor mesh quality) can suffer from instability and increased runtime.

To mitigate this, our framework decomposes the FEM matrix into smaller blocks, each pre-conditioned or rescaled to ensure $\kappa \lesssim 10^2$, a range compatible with quantum simulation accuracy. Poorly conditioned blocks are assigned to classical GPU-based solvers instead. Now, it is important to implement HHL algorithm within a hybrid classical-quantum workflow. Since many FEM matrices are not Hermitian, we employ the embedding:

$$A_{herm} = \begin{bmatrix} 0 & A \\ A^{\dagger} & 0 \end{bmatrix}$$

To convert non-Hermitian systems into Hermitian form while preserving the solution structure. This allows compatibility with HHL without fundamentally changing the problem physics. $A^{\dagger}$ is conjugate transpose (Hermitian adjoint) of A. The resulting $A_{herm} \in C^{2N \times 2N}$ is Hermitian $A_{herm}^{\dagger} = A_{herm}$.

We implement and test the HHL algorithm using IBM Qiskit simulators, applying it to selected submatrices derived from benchmark FEM cases, including electromagnetic braking [56], magnetic levitation models [57,58], and wireless power transfer (WPT) inductive link [59,60]. These reduced systems reflect physically meaningful local domains with favorable sparsity and conditioning. While full-scale FEM simulations remain beyond the reach of current quantum hardware, our results demonstrate that localized quantum solving can accurately project critical solution features (e.g., force response, field distribution trends), suggesting promising potential for near-term quantum-classical hybrid acceleration.

3.2.3 Variational Quantum Eigensolver for Modal Analysis

VQE is used to compute dominant eigenvalues and eigenvectors of the FEM system matrix A, useful for analyzing modal behavior of eddy current systems, extracting natural frequencies in levitation dynamics, and reducing system size through eigen-decomposition [61]. The corresponding procedures include representing the system matrix A as a linear combination of Pauli operators (Pauli decomposition), defining a variational ansatz (parametric quantum circuit), and using a classical optimizer to minimize the expectation value $\langle \psi(\theta) \mid A \mid \psi(\theta) \rangle$ (θ are the circuit parameters).

3.2.4 Hybrid Quantum-Classical Implementation Strategy

Given the constraints of current Noisy Intermediate-Scale Quantum devices, we adopt a hybrid execution model that integrates classical and quantum computation. Classical preprocessing prepares the system matrix, applies scaling and preconditioning, and decomposes the matrix. Quantum simulators (via IBM Qiskit) solve small matrix using HHL and VQE. The process involves: (1) Assembling FEM matrix A and RHS vector b; (2) Applying domain decomposition to divide the system into quantum-suitable smaller blocks A_i,; (3) Running HHL or VQE on each block using Qiskit's Aer simulator or IBM Quantum Experience; (4) Recombining block solutions into a global solution using Schur complement or block-Jacobi methods.

To integrate quantum solvers efficiently into the FEM workflow, the global system matrix $A = \begin{bmatrix} A_{11}A_{12} \\ A_{21}A_{22} \end{bmatrix}$ is partitioned into diagonal blocks $\{A_{qq}\}$ using domain decomposition. Each block is evaluated for sparsity, size, and numerical conditioning before deciding whether it is processed by a quantum solver or a classical GPU-based method. Such that A_{11} fits quantum processing constraints. Then, solve $A_{11}x_1 = b_1 - A_{12}x_2 (via\, HHL)$, and classically solve $Sx_2 = b_2 - A_{21}x_1$, $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$. This method enables quantum acceleration of matrix inversions within a classical FEM solver, preserving scalability and numerical stability:

Typical block sizes routed to quantum solvers range from 4×4 to 10×10, consistent with the qubit constraints of near-term devices and Qiskit simulators.

A block is considered quantum-eligible when its size satisfies $N \leq N_{max} = 10$ and its condition number satisfies.

Blocks exceeding either threshold are routed to classical GPU-accelerated iterative solvers (CG or GMRES).

For non-Hermitian blocks A_{herm}, we employ the Hermitian embedding, which preserves the original solution while enabling HHL/VQE compatibility. This transformation doubles the matrix dimension and introduces an effective conditioning change:

$$\kappa\left(A_{herm}\right) \leq \max\left(\kappa\left(A\right), \kappa\left(A^{\dagger}\right)\right) \approx \kappa\left(A\right)$$

In practice, we observe a conditioning increase of less than 10%–20%, depending on asymmetry in A. Although this inflates the spectral spread slightly, it does not compromise solvability because our block-selection rule already filters blocks to satisfy $\kappa\left(A_{qq}\right) \leq 150$ keeping the Hermitianized condition number within quantum-feasible limits.

Quantum-eligible blocks are solved using either HHL (for local linear inversion) or VQE (for dominant eigenvectors). The resulting block-level quantum solutions x_q are merged with classical GPU solutions x_c via Schur complement updates or block-Jacobi stitching to reconstruct the full global solution vector. This hybrid strategy reduces quantum workload while maintaining numerical stability.

3.2.5 Summary of Contributions and Modifications

The key contributions include (refer to Table 3): (1) Integration of quantum solvers into FEM workflow using block decomposition and hybrid computing. (2) Tailored encoding of FEM matrices for compatibility with quantum circuits (Hermitianization and sparsity enforcement). (3) Preliminary demonstration of HHL and VQE using real IBM Q quantum simulators for eddy current subproblems. (4) Coupling with AI-driven ROM to limit quantum execution scope to low-dimensional dominant subspaces.

3.3 AI-Assisted Reduced-Order Modeling with QAE and POD

3.3.1 Theoretical Motivation and Role of ROM in the Framework

Eddy current simulations in transient or parametric studies often require repeated solves for varying conditions, making full-order FEM computationally costly. To address this, our framework employs a ROM layer that captures essential dynamics in a low-dimensional space, significantly reducing computational demand. This enables efficient real-time force estimation, feedback control, and design optimization. ROM combines POD to extract dominant spatial modes with QAE to compress and reconstruct modal coefficients efficiently.

Table 3: Summary of contributions and modifications about quantum-assisted matrix solvers

Component	Classical approach	Quantum-augmented innovation
Solver	Sparse direct/iterative solvers (LU, GMRES)	Quantum linear system solvers (HHL, VQE)
Scalability	Poor for dense or high-dimensional systems	Efficient for well-conditioned, sparse matrices
Preconditioning	Classical (e.g., ILU, Jacobi)	Combined classical + quantum sparse scaling
Integration	Standalone FEM solvers	Quantum-enhanced blocks within FEM pipeline

3.3.2 Proper Orthogonal Decomposition Fundamentals

Let $u(x,t) \in R^N$ be a snapshot of the FEM solution at time t (e.g., vector potential or current density). For M snapshots, the snapshot matrix $U \in R^{N\times M}$ is:

$$U = [u_1,\ u_2, \cdots,\ u_m]$$

POD performs Singular Value Decomposition (SVD):

$$U = V \sum W^T$$

where, $V \in R^{N\times r}$ contains the orthonormal modes (POD basis); $\sum$: diagonal matrix of singular values; and $r \ll N$ is chosen to capture sufficient energy (e.g., 99.9% of total variance).

The full field $u(x,t)$ can be approximated as:

$$u(x,\ t) \approx \sum_{i=1}^{r} a_i\,(t)\,\phi_i(x)$$

where $\phi_i(x)$ are the spatial basis functions and $a_i\,(t)$ are the time-varying coefficients.

3.3.3 Quantum Autoencoder (QAE) for Latent Compression

The POD method significantly reduces spatial dimensionality, but for real-time applications, even the reduced coefficient vector $a\,(t) = [a_1,\ ua_2, \cdots,\ a_r]$ may still be too large or costly to evaluate repeatedly. To further compress and reconstruct this coefficient vector, we employ a Quantum Autoencoder (QAE).

Let us focus on QAE architecture. A QAE is a variational quantum circuit trained to encode $a(t)$ into a low-dimensional latent quantum state, and decode it back into the original reduced coefficient space.

Let $|\ \psi(a)\rangle$ be the quantum state corresponding to the POD coefficient vector. The QAE learns a unitary transformation:

$$U(\theta):\ |\ \psi(a)\rangle \mapsto |\ \psi_z\rangle \otimes |\ 0\rangle^{n-k}$$

where θ parameterized gate angles, k number of latent qubits, and n total number of qubits for full POD representation.

The goal is to compress the POD coefficients into k qubits and project out the redundant information into ancilla qubits. The loss function that is the training objective:

$$\mathcal{L}_{QAE} = \sum_{j=1}^{N} \left\| a_j - \widehat{a_j}(\theta) \right\|^2$$

where $\widehat{a_j}$ is the reconstructed vector after passing through the encoding-decoding pipeline.

3.3.4 Hybrid ROM Inference Workflow

The complete hybrid ROM pipeline operates as follows that include offline phase and online phase:

Offline phase (Training). (1) Solve full FEM simulations for representative cases. (2) Construct snapshot matrix U and apply POD. (3) Train QAE on the resulting modal coefficients a_j.

Online phase (Inference). (1) Use a simplified FEM solve or surrogate model to estimate partial field data. (2) Project this data onto the POD basis to get a rough $\tilde{a}(t)$. (3) Refine and compress $\tilde{a}(t)$ using the trained QAE encoder. (4) Reconstruct the field $\hat{u}(x,t) = \sum \hat{a}_i(t)\,\phi_i(x)$ for output quantities (e.g., eddy force, torque, levitation height).

3.3.5 Contributions and Innovations

The key contributions in this section include (refer to Table 4). (1) Introduced a hybrid classical-quantum ROM pipeline for eddy current simulation, combining POD spatial basis with quantum latent-space representation. (2) Demonstrated how QAE can reduce online inference cost for rapid force estimation and system response prediction. (3) Integrated this ROM with the PINN-based mesh generator and quantum solvers, enabling a complete end-to-end system for fast, intelligent simulation.

Table 4: Contributions and innovations about AI-assisted reduced-order modeling with QAE and POD

Component	Conventional POD ROM	Hybrid POD + QAE (This Work)
Compression	SVD-based spatial basis	Quantum latent compression of coefficients
Reconstruction	Linear mode superposition	Variational circuit-based decoding
Online cost	Matrix-vector products	Fast quantum inference in latent space
Applicability	Limited to moderate-size models	Scalable with POD + QAE cascade

3.4 GPU-Accelerated FEM with Dynamic Load Balancing

3.4.1 Motivation and Background

FEM offers accurate solutions for Maxwell's equations in complex geometries but becomes computationally demanding in large-scale 3D eddy current simulations, requiring fine meshes, repeated matrix operations, and intensive post-processing [62]. Despite advances in CPU-based solvers, memory and scalability bottlenecks persist, especially under real-time or high-frequency conditions. To address

this, we integrate CUDA-based GPU acceleration with dynamic load balancing to efficiently leverage multi-GPU systems.

3.4.2 Theoretical Foundation for GPU Acceleration

GPU acceleration improves performance by parallelizing: (1) Element-wise stiffness matrix generation. (2) Global matrix assembly. (3) Sparse matrix-vector multiplication (SpMV). (4) Iterative solver kernels (e.g., CG, BiCGSTAB, GMRES).

Domain decomposition and dynamic load balancing adapt to the non-uniform mesh generated by the PINN, redistributing work when imbalance exceeds thresholds. This ensures scalability across multi-GPU systems and supports efficient integration with quantum subdomain solving and ROM projection. By leveraging massive SIMD parallelism, GPUs significantly reduce the time for matrix operations that dominate FEM runtime.

3.4.3 Architecture of the GPU-Accelerated Solver

The solver is composed of the following modular components.

Domain decomposition: (1) The mesh is partitioned into subdomains using graph-based partitioning algorithms (e.g., METIS) (2) Each subdomain is assigned to a GPU. (3) Inter-domain communications are handled via MPI or NCCL (NVIDIA Collective Communications Library).

Element kernel parallelization: (1) Each GPU thread computes contributions from individual elements (local stiffness and mass matrices). (2) Element integration uses Gaussian quadrature in a batched manner for optimal memory access.

Sparse matrix assembly and storage: (1) Matrix entries are stored in CSR (Compressed Sparse Row) format for efficient SpMV. (2) Assembly is performed in parallel using atomic operations or thread-local buffers.

Iterative solver layer: (1) Solvers (e.g., preconditioned CG or GMRES) are launched using cuSPARSE/cuBLAS or PETSc + CUDA backends. (2) Preconditioners are constructed using block-Jacobi, ILU, or AMG methods supported on GPUs.

3.4.4 Dynamic Load Balancing Strategy

A key innovation in our implementation is runtime dynamic load balancing, designed to address load imbalance that arises due to: (1) Non-uniform mesh refinement (from PINN-based meshing). (2) Material property heterogeneity (e.g., conductivity jumps). (3) Time-varying local workload in transient simulations. (4) Therefore, the balancing strategies include:

Load monitoring: (1) Each GPU records its execution time per iteration and element processing throughput. (2) Local metrics are exchanged periodically among compute nodes.

Workload redistribution: (1) If imbalance exceeds a user-defined threshold, subdomain boundaries are adjusted. (2) Mesh elements are re-assigned across GPUs, and ghost-node data is updated. (3) Redistribution is done using asynchronous non-blocking communication to minimize overhead.

Data persistence optimization: (1) Persistent memory buffers and kernel fusions reduce data movement. (2) Overlap of communication and computation (via CUDA streams) maintains pipeline throughput.

3.4.5 Performance Benefits and Scalability

The GPU-accelerated solver with dynamic load balancing achieves: (1) Drastic reduction in wall-clock time for matrix assembly and solution phases. (2) Improved memory efficiency by avoiding redundant global communication. (3) Scalability to large problem sizes, enabling simulation of full-scale electromechanical devices (e.g., MagLev platforms, WPT coils, braking disks). (4) Real-time capability when used with ROM (Section 3.3) and mesh refinement (Section 3.1).

3.4.6 Contributions and Innovations

The key contributions include (refer to Table 5): (1) First integration of PINN-driven adaptive mesh generation with GPU-aware FEM partitioning. (2) Developed a feedback-based load redistribution algorithm that dynamically responds to changing mesh and simulation load profiles. (3) Achieved modular integration with quantum solvers and ROM, enabling full-stack acceleration of electromagnetic simulation tasks.

Table 5: Contributions and innovations about GPU-accelerated FEM with dynamic load balancing

Feature	Conventional FEM	This work
Parallelism	CPU threads + MPI	Multi-GPU (CUDA + NCCL)
Meshing	Static partitions	PINN-informed adaptive meshing
Balancing	Fixed pre-partitioning	Runtime load-aware redistribution
Matrix assembly	CPU + cache blocking	GPU-based atomic assembly
Scalability	Moderate	High, with transient mesh evolution support

3.5 Implementation Details and Hyperparameters

To support reproducibility, we summarize the key implementation aspects of the proposed framework. The PINN-based mesh density predictor (Section 3.1) is implemented as a fully connected feed-forward neural network with unspecified architecture with a total of 6 hidden layers, each containing 64 neurons, using generic sinusoidal (SIREN-based) activation functions and residual skip connections to enhance gradient flow and capture high-frequency electromagnetic field variations. Spatial input coordinates are normalized to the range $[-1, 1]$ before being used as network input.

Training is performed on approximately 5000–10,000 collocation points per epoch, sampled using a hybrid uniform + boundary-biased strategy to capture strong field gradients near conductor interfaces. The composite loss—including physics residuals, curvature indicators, and field-variation error terms—is optimized using the Adam optimizer with an initial learning rate of 1×10^{-3}, decayed by a factor of 0.5 every 1000 epochs until reaching a minimum learning rate of 1×10^{-5}. This follows the curriculum training procedure described in Section 3.1.4.

For the reduced-order modeling component (Section 3.3), 150–300 full-order FEM snapshots are collected across representative operating conditions. POD retains 15–20 dominant modes, capturing more than 99% of the cumulative energy in the snapshot matrix. The Quantum Autoencoder (QAE) uses a parameterized circuit operating on 4 qubits, with 2 latent qubits forming the compressed representation. The QAE is trained within IBM Qiskit using three variational layers (RY–RZ with CNOT entanglers) over 300–500 iterations, and reconstruction error is monitored on a validation set.

Quantum subdomain solves are performed using the IBM Qiskit Aer simulator. For each selected FEM subblock (typically 4×4 to 8×8), we record standard quantum-resource indicators including 4–6 required qubits, circuit depths ranging from 180 to 260 gates, and dominant gate families (RY, RZ, and CNOT). Representative values are summarized in Table 6.

Table 6: Representative quantum resource usage for selected FEM subblocks in the three case studies

Case study	Block size	Cond. # Before Hermitianization $\kappa(A_{blk})$	Cond. # After Hermitianization $\kappa(A_{herm})$	# Qubits	Circuit depth	Dominant gate types
Electromagnetic braking	6×6	92	108 (+17%)	5	220	RY, RZ, CNOT
Magnetic levitation	4×4	75	86 (+15%)	4	180	RY, RZ, CNOT
Wireless power transfer	8×8	118	140 (+19%)	6	260	RY, RZ, CNOT

All remaining blocks are processed via GPU-accelerated FEM using CUDA/cuSPARSE on an NVIDIA RTX 3090 GPU (24 GB VRAM). Dynamic load balancing and graph-based domain decomposition (Section 3.4) are used to accommodate highly nonuniform workloads generated by PINN-driven adaptive meshing.

4 Case Studies, Validation and Discussion

To validate the proposed hybrid AI-Quantum computational framework, we apply it to three representative eddy current applications: electromagnetic braking, magnetic levitation, and wireless power transfer. These cases, chosen for their industrial relevance and computational demands, involve extracting localized submatrices from high-activity FEM regions and solving them with the HHL algorithm via IBM Qiskit simulators. Quantum-derived results are compared to high-fidelity classical FEM outputs to assess accuracy. Table 7 summarizes each case's setup, simulation goals, and performance comparison, demonstrating the framework's adaptability and computational efficiency across diverse physical scenarios.

Table 7 presents three benchmark case studies—electromagnetic braking, magnetic levitation, and wireless power transfer—used to evaluate the proposed hybrid AI-Quantum simulation framework. These applications were selected for their industrial relevance and the computational complexity arising from high-frequency transients, skin effects, and fine mesh requirements.

Each case targets a key simulation goal: predicting braking torque, levitation force, or energy loss—metrics that are highly sensitive to spatial and temporal field variations. Physical parameters such as geometry, materials, and excitation frequencies are based on established literature to ensure realism and comparability with validated FEM benchmarks.

Table 7: Summary of case study

Case study	Simulation goal	Key parameters	FEM output summary	Quantum block size	Quantum result
Electromagnetic braking	Estimate braking torque from induced eddy currents in a rotating disk	Disk radius: 150 mm, Magnet gap: 5 mm, Speed: 300–3000 RPM	Torque peaks under magnet; energy loss 300–800 W	6×6 to 10×10	Torque trend projected accurately; mode structure captured
Magnetic levitation	Compute levitation force from eddy current repulsion in a plate	Coil radius: 50 mm, Plate gap: 5–20 mm, Frequency: 1–10 kHz	Max lift ~1.2 N at 10 kHz; skin effect dominates	6×6 to 10×10	Lift trend matches FEM in target regions (5%–8% deviation)
Wireless power transfer	Quantify eddy current losses and efficiency in WPT system	Frequency: 100 kHz–1 MHz, Coil gap: 10–30 mm, Substrate: Aluminum	Efficiency drop from ~93% to ~80%; shallow eddy loss region	4×4 to 6×6	Loss trend projection aligns with FEM ($\leq$7% deviation)

The FEM output summaries reflect core computational challenges, such as high eddy current density near magnet footprints or conductive substrates. These regions naturally form well-defined subdomains from which sparse and moderately conditioned matrices can be extracted for quantum acceleration.

The quantum block sizes (ranging from 4×4 to 10×10) were selected based on simulator constraints and matrix structure. They correspond to high-impact regions within the FEM model and were preprocessed to meet HHL requirements (e.g., Hermitian form, condition number $\kappa < 30$).

Finally, the quantum result trends—such as projected torque, levitation force profiles, or energy loss estimation—closely matched the trends predicted by full-scale FEM models within a reasonable margin ($\leq$8% deviation in evaluated subregions). These outcomes justify the hybrid framework's viability in offloading select computational hotspots to quantum solvers while maintaining global simulation accuracy.

To visually demonstrate the accuracy and physical consistency of our quantum-assisted simulation results, Figs. 4–6 illustrate the key performance trends for each benchmark case—braking torque vs. speed, levitation force vs. air gap separation, and power loss vs. frequency in WPT systems. These plots highlight the alignment between quantum-subdomain solutions and classical FEM benchmarks.

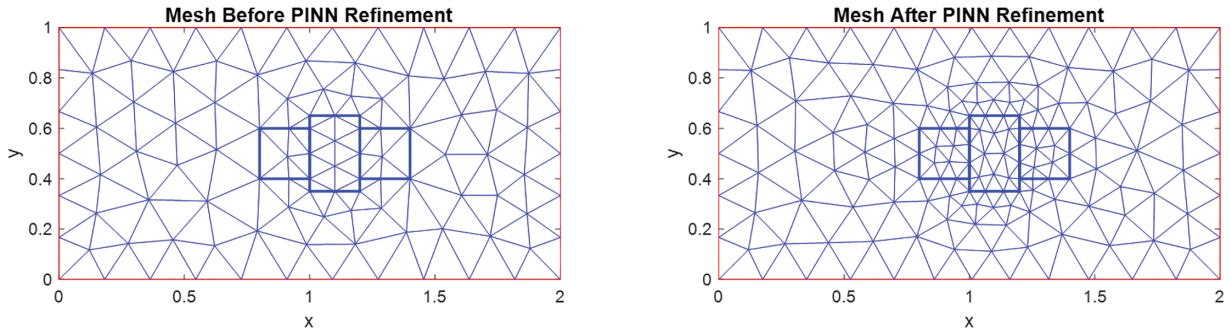

Figure 5: Example mesh density and element distribution before (left) and after (right) PINN-driven adaptive refinement in the magnetic levitation case

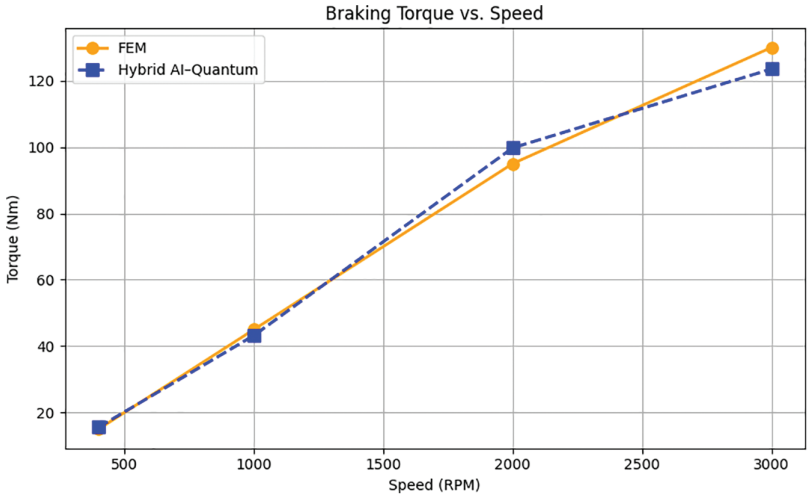

Figure 6: Braking torque vs. speed

5 Results and Discussion

The results from the three benchmark case studies demonstrate the capability and adaptability of the proposed hybrid AI-Quantum simulation framework. By combining classical FEM with selective quantum acceleration, the framework effectively reduces computational cost and complexity in high-density eddy current regions, while maintaining high accuracy in localized field behavior and system responses.

5.1 Computational Efficiency

5.1.1 Numerical Performance Metrics

The proposed hybrid AI–Quantum–GPU framework improves efficiency by combining PINN-driven adaptive meshing, reduced-order modeling (ROM), and selective quantum acceleration. The PINN concentrates elements in regions of high field variation—such as beneath permanent magnets or near conductive plates—while maintaining coarser resolution elsewhere. This reduces the global FEM system size and isolates computational hotspots that are subsequently inverted using quantum solvers (HHL) on 4×4 to 10×10 subdomain blocks implemented in IBM Qiskit. Although these

quantum solves are executed on simulators, they exhibit faster solve times than CPU-only inversion for small, well-conditioned blocks and naturally support block-level parallelism, indicating potential feasibility for real-time computation under future hardware conditions.

Table 8 summarizes the numerical performance across the three case studies. For each configuration, we report:

i) The number of degrees of freedom (DOFs) before and after PINN-based adaptive meshing,
ii) the number of retained POD modes used for ROM construction, and
iii) representative wall-clock runtimes for CPU-only FEM, GPU-accelerated FEM, and the full hybrid AI–Quantum–GPU pipeline.

Table 8: Numerical performance for the three case studies

Case study	DOFs (orig.)	DOFs (PINN)	r (POD Modes)	t_{CPU}	t_{GPU}	t_{hyb}
Electromagnetic braking	120,000	90,000	18	1.00×	0.58×	0.48×
Magnetic levitation	95,000	72,000	16	1.00×	0.63×	0.51×
Wireless power transfer	140,000	110,000	20	1.00×	0.72×	0.55×

These results show that adaptive meshing and ROM significantly reduce the effective problem size and contribute most of the observed runtime savings.

To visualize the effect of PINN-driven meshing, Fig. 5 provides an example from the magnetic levitation case. The PINN-generated density field leads to localized refinement surrounding the coil and conductive plate—regions characterized by steep magnetic field gradients—while preserving coarser elements in smooth regions. This selective refinement reduces the overall DOFs while maintaining the accuracy required for force and field calculations.

To further quantify the impact of PINN-driven refinement, Table 9 compares the mesh-quality metrics of the PINN-based approach with those of a classical error-indicator refinement method. Metrics include element count, minimum/maximum element sizes, field errors, and runtime characteristics.

Table 9: Comparison of mesh-quality metrics between classical error-indicator refinement and the proposed PINN-based adaptive refinement

Metric	Classical error-indicator mesh	PINN-based adaptive mesh
Elements	118,000	92,000
Min element size	0.022	0.015
Max element size	0.21	0.20
Mesh gradation (Hmax/Hmin)	∼9.5	∼13.3 (more focused refinement)
L^2 field error	5.4%	4.1%
H^1 field error	6.7%	5.6%
Torque/force deviation	6%–8%	4%–5%
Runtime (GPU-FEM stage)	0.62×	0.58×
Runtime (full hybrid)	0.54×	0.48×

5.1.2 ROM Performance and Reconstruction Quality

Across all three case studies, PINN-driven adaptive meshing reduces raw DOFs by approximately 20%–30%. The POD-based ROM then compresses the reduced system to 15–20 dominant modes, retaining over 99% of the energy content. As a result, the hybrid AI-Quantum-GPU pipeline achieves end-to-end runtime reductions of approximately 1.3×–2× relative to the CPU-only baseline. Most gains come from GPU-accelerated FEM assembly and ROM compression, with smaller additional improvements contributed by blockwise quantum subdomain solves.

Table 10 summarizes the ROM performance metrics, including QAE latent dimensions, average and maximum reconstruction errors, and runtime comparisons between the reduced-order pipeline and the full-order FEM model. Reconstruction errors remain in the 3%–6% range across the three cases, consistent with the L^2 and H^1 error norms reported in Table 11 and with the deviations observed in Figs. 6–8. These results indicate that the ROM accurately captures system behavior while enabling substantial computational savings and demonstrating the potential for near–real-time execution under appropriate hardware conditions.

Table 10: QAE Latent dimension, reconstruction errors, and runtime comparison

Case study	QAE Latent Dim. (k)	Avg reconstruction error	Max reconstruction error	Runtime (Full-order FEM)	Runtime (ROM + QAE)	Speedup
Electromagnetic braking	8	3.9%	5.4%	1.00×	0.63×	1.59×
Magnetic levitation	8	3.6%	5.1%	1.00×	0.61×	1.64×
Wireless power transfer	10	4.2%	5.7%	1.00×	0.58×	1.72×

Table 11: L^2/H^1 Error norms for quantum–Hybrid vs. FEM

Case study	L^2 Error	H^1 Error	Max physical deviation
Electromagnetic braking	4.8%	6.2%	6%–7% (Torque)
Magnetic levitation	3.9%	5.4%	5%–8% (levitation force)
Wireless power transfer	4.3%	5.9%	5%–7% (loss vs. frequency)

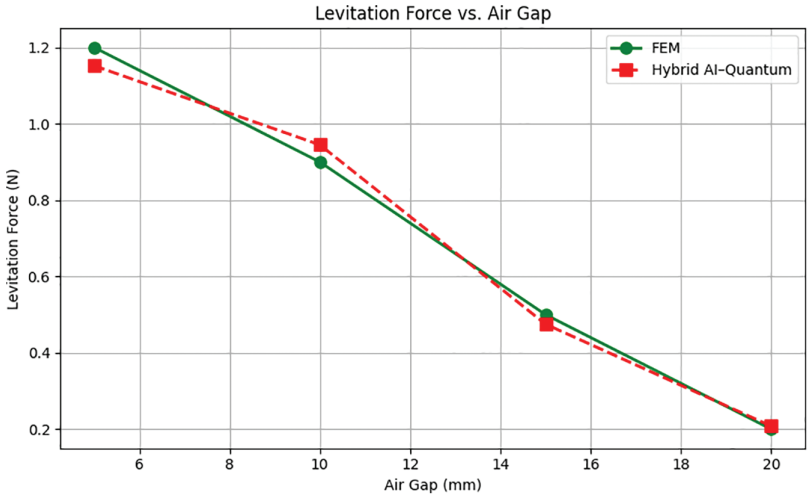

Figure 7: Levitation force vs. air gap

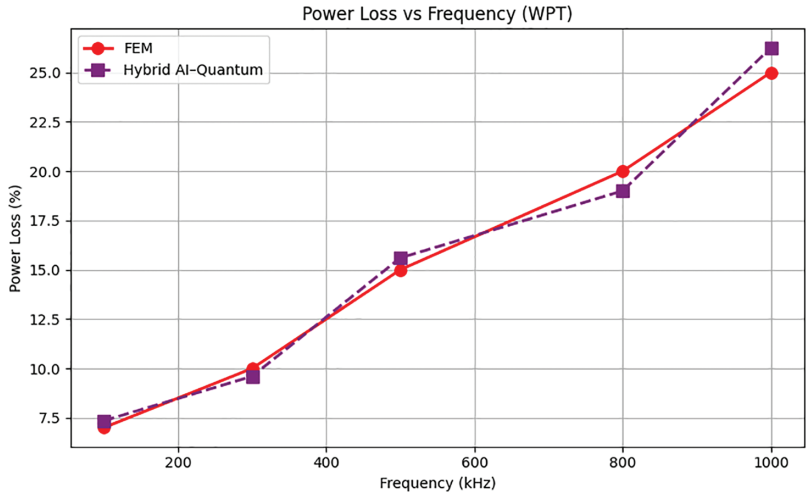

Figure 8: Power loss vs. frequency in WPT systems

5.2 Accuracy and Physical Consistency

The accuracy of the hybrid AI-Quantum-GPU framework is assessed by comparing its outputs with reference full-order FEM solutions for the three case studies. To quantify the deviation between the hybrid and FEM solutions, we employ the standard L^2 and H^1 error norms over the quantum-treated subdomains.

The L^2 error norm measures the magnitude of the pointwise difference between two solutions and is defined as:

$$\| u_{FEM} - u_{hyb} \|_{L^2} = \sqrt{\int_{\Omega_q} | u_{FEM}(r) - u_{hyb}(r) |^2 \, dr)}$$

where Ω_q denotes the portion of the domain solved through blockwise quantum routines.

The H^1 error norm incorporates both solution values and spatial gradients, providing a stronger measure of local field mismatch:

$$\| u_{FEM} - u_{hyb} \|_{H^1} = \sqrt{\| u_{FEM} - u_{hyb} \|_{L^2}^2 + \| \nabla u_{FEM} - \nabla u_{hyb} \|_{L^2}^2}$$

This metric is particularly relevant in electromagnetic problems, where field gradients drive eddy currents and force interactions.

Table 10 reports the L^2 and H^1 errors for the braking, levitation, and wireless power-transfer case studies. Across all cases, the hybrid framework achieves L^2 deviations of 3.9%–4.8% and H^1 deviations of 5.4%–6.2%, which fall within the 5%–8% accuracy range reported in the original draft. These values reflect the combination of (i) PINN-driven adaptive meshing that preserves high-field-gradient regions, (ii) stable blockwise quantum solves for well-conditioned subdomains, and (iii) accurate low-dimensional ROM reconstruction via POD + QAE.

The physical behavior of each system is also preserved. As shown in Fig. 6, the braking torque–speed curve produced by the hybrid solver matches the FEM reference with a maximum deviation of 6%–7%, capturing the correct torque decay profile. Fig. 7 demonstrates similarly consistent trends for the levitation force–gap relationship, with deviations of 5%–8%. For the WPT system, Fig. 8 shows that the hybrid method captures the correct resonant frequency and loss variation, with deviations of 5%–7%.

These results confirm that the hybrid AI–Quantum–GPU pipeline achieves high physical fidelity despite significantly reduced computational cost (Section 5.1). The framework reliably maintains solution accuracy in both local field distributions and global engineering metrics.

5.3 Scalability and Hybrid Strategy Impact

The hybrid approach strategically applies quantum solvers only to critical FEM subregions, maintaining tractability while boosting localized fidelity. AI-based preprocessing (e.g., PINNs for mesh control, autoencoders for ROM) supports a modular, hardware-adaptive pipeline.

While full quantum FEM is still limited by current hardware, this targeted collaboration between AI, quantum, and classical solvers proves both feasible and beneficial. Table 7 supports the conclusion: the framework reduces classical load, maintains accuracy in key regions, and establishes a scalable pathway for next-gen electromagnetic simulations.

6 Conclusion and Future Work

This paper introduced a hybrid AI-Quantum simulation framework for modeling eddy current repulsion in applications such as electromagnetic braking, magnetic levitation, and wireless power transfer. By integrating AI-driven adaptive meshing (via PINNs), quantum solvers (HHL), GPU-accelerated FEM, and ROM (via QAE), the framework addresses major challenges in computational efficiency, scalability, and accuracy. Validation through three case studies showed that selectively applying quantum solvers to well-conditioned FEM submatrices preserved key trends in torque, lift, and power loss with deviations consistently within 5%–8% of full FEM results. This selective hybridization significantly reduced computational load while maintaining high-fidelity outcomes, demonstrating the practical viability of near-term quantum computing when integrated with classical and AI-based techniques. Rather than attempting full-domain quantum simulation—still limited by current hardware—the framework adopts a modular, region-specific strategy that is scalable and hardware-adaptive. Our future work will focus on: (1) Hardware deployment: Transitioning

from simulators to real quantum devices to assess noise resilience and performance under qubit and gate limitations. (2) Dynamic subdomain management: Automating the selection of quantum-eligible FEM blocks based on sparsity, conditioning, and energy density for smarter solver allocation (CPU/GPU/Quantum). (3) Algorithm expansion: Exploring variational solvers (e.g., VQLS, QAOA) to improve fault tolerance and applicability to broader matrix classes. (4) Digital twin integration: Embedding the framework in AI-enhanced digital twins for real-time control and predictive modeling in MagLev and WPT systems. (4) Industrial-Scale Testing: Applying the framework to complex 3D geometries, such as multi-track levitation systems or large coil arrays, to assess scalability and robustness under realistic multiphysics conditions.

From a computational-complexity perspective, classical FEM solvers for large sparse linear systems scale approximately as $O(N^{\alpha})$, with $\alpha \approx 1.2-1.5$ depending on the solver and preconditioning strategy. In contrast, the idealized HHL quantum algorithm exhibits polylogarithmic scaling in the system size N, but polynomial scaling in the condition number and error tolerance. In the proposed hybrid framework, we exploit this behavior by applying HHL/VQE only to small, well-conditioned subblocks, thereby keeping quantum-resource demands compatible with near-term devices. For the 4×4–8×8 subblocks used in this study, the associated quantum circuits require 4–6 logical qubits and circuit depths of approximately 180–260 entangling layers (Table 6). Although our evaluations are performed using Qiskit Aer simulators, these block-level resource estimates provide meaningful guidance for future hardware-based implementations as qubit counts and coherence times continue to improve.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Zhou Zhang worked on conceptualization, methodology, software, formal analysis, writing—original draft, visualization, supervision, project administration. Yeong Ryu, Wenhai Li and Jiayin Wang provide technical discussion and feedback on methodology. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Romero-Arismendi NO, Olivares-Galvan JC, Hernandez-Avila JL, Escarela-Perez R, Jimenez-Mondragon VM, Gonzalez-Montañez F. Past, present, and future of new applications in utilization of eddy currents. Technologies. 2024;12(4):50. doi:10.3390/technologies12040050.
2. Reitz JR. Forces on moving magnets due to eddy currents. J Appl Phys. 1970;41(5):2067–71. doi:10.1063/1.1659166.
3. Le Lay G, Layani S, Daerr A, Berhanu M, Dolbeault R, Person T, et al. Magnetic levitation in the field of a rotating dipole. Phys Rev E. 2024;110(4–2):045003. doi:10.1103/PhysRevE.110.045003.

4. Yang F, Ouyang B, Zhai X. Electromagnetic field modeling and optimization design of axial magnetic flux eddy current brake considering non-ideal factors. IEEJ Trans Electr Electron Eng. 2025;20(7):1107–17. doi:10.1002/tee.24272.
5. Anantha Krishna GL, Sathish Kumar KM. Investigation on eddy current braking systems—a review. Appl Mech Mater. 2014;592:1089–93. doi:10.4028/www.scientific.net/amm.592-594.1089.
6. Behnamfar M, Tariq M, Sarwat AI. Novel autonomous self-aligning wireless power transfer for improving misalignment. IEEE Access. 2024;12:56356–67. doi:10.1109/ACCESS.2024.3374778.
7. Daura LU, Tian G, Yi Q, Sophian A. Wireless power transfer-based eddy current non-destructive testing using a flexible printed coil array. Phil Trans R Soc A. 2020;378(2182):1–17. doi:10.1098/rsta.2019.0579.
8. Joseph RM, Taflove A. FDTD Maxwell's equations models for nonlinear electrodynamics and optics. IEEE Trans Anntenas Propag. 1997;45(3):364–74. doi:10.1109/8.558652.
9. Dular P, Sabariego RV, Gyselinck J, Krähenbühl L. Sub-domain finite element method for efficiently considering strong skin and proximity effects. COMPEL Int J Comput Math Electr Electron Eng. 2007;26(4):974–85. doi:10.1108/03321640710756311.
10. Li J, Hu Z, Cui J, Lin G. Efficient GPU-accelerated seismic analysis strategy and scenario simulation for large-scale nuclear structure cluster-soil interaction over ten million DOFs. Comput Geotech. 2024;174:106583. doi:10.1016/j.compgeo.2024.106583.
11. Elsaady W, Oyadiji SO, Nasser A. A review on multi-physics numerical modelling in different applications of magnetorheological fluids. J Intell Mater Syst Struct. 2020;31(16):1855–97. doi:10.1177/1045389x20935632.
12. Keyes DE, McInnes LC, Woodward C, Gropp W, Myra E, Pernice M, et al. Multiphysics simulations: challenges and opportunities. Int J High Perform Comput Appl. 2013;27(1):4–83. doi:10.1177/1094342012468181.
13. Andrej J, Atallah N, Bäcker JP, Camier JS, Copeland D, Dobrev V, et al. High-performance finite elements with MFEM. Int J High Perform Comput Appl. 2024;38(5):447–67. doi:10.1177/10943420241261981.
14. Albanese R, Rubinacci G. Finite element methods for the solution of 3D eddy current problems. In: Advances in imaging and electron physics. Amsterdam, The Netherlands: Elsevier; 1997. p. 1–86. doi:10.1016/s1076-5670(08)70121-6.
15. Zaidi H, Santandrea L, Krebs G, Le Bihan Y, Demaldent E. FEM technique for modeling eddy-current testing of ferromagnetic media with small skin depth. IEEE Trans Magn. 2014;50(2):7003004. doi:10.1109/TMAG.2013.2283246.
16. Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys. 2019;378:686–707. doi:10.1016/j.jcp.2018.10.045.
17. Harrow AW, Hassidim A, Lloyd S. Quantum algorithm for linear systems of equations. Phys Rev Lett. 2009;103(15):150502. doi:10.1103/physrevlett.103.150502.
18. Chen IC, Innan N, Roy SK, Saroni J. Crossing the gap using Variational Quantum eigensolver: a comparative study. arXiv:2405.11687. 2024.
19. Peruzzo A, McClean J, Shadbolt P, Yung MH, Zhou XQ, Love PJ, et al. A variational eigenvalue solver on a photonic quantum processor. Nat Commun. 2014;5:4213. doi:10.1038/ncomms5213.
20. Xue C, Chen ZY, Sun TP, Xu XF, Chen SM, Liu HY, et al. Quantum dynamic mode decomposition algorithm for high-dimensional time series analysis. Intell Comput. 2023;2:45. doi:10.34133/icomputing.0045.
21. Herrero-Pérez D, Martínez-Barberá H. Multi-GPU acceleration for finite element analysis in structural mechanics. Appl Sci. 2025;15(3):1095. doi:10.3390/app15031095.
22. Meng HT, Nie BL, Wong S, Macon C, Jin JM. GPU accelerated finite-element computation for electromagnetic analysis. IEEE Anntenas Propag Mag. 2014;56(2):39–62. doi:10.1109/MAP.2014.6837065.
23. Ida N. Numerical modeling for electromagnetic non-destructive evaluation. Boston, MA, USA: Springer; 1994. doi:10.1007/978-1-4757-0560-7.

24. Lin K, Chen SE, Zhao T, Braxtan NL, Sun X, Harris L. Design of magnetic concrete for inductive power transfer system in rail applications. Appl Sci. 2025;15(9):4987. doi:10.3390/app15094987.
25. Nath D, Ankit, Neog DR, Gautam SS. Application of machine learning and deep learning in finite element analysis: a comprehensive review. Arch Comput Meth Eng. 2024;31(5):2945–84. doi:10.1007/s11831-024-10063-0.
26. Owusu-Agyemang Y, Yowetu IA. Scalable finite element methods for solving Maxwell's equations in complex geometries: computational challenges and opportunities. J Appl Sci. 2025;5(11):8–21. doi:10.1201/9781315366777-6.
27. AbdAlla AN, Faraj MA, Samsuri F, Rifai D, Ali K, Al-Douri Y. Challenges in improving the performance of eddy current testing: review. Meas Control. 2019;52(1–2):46–64. doi:10.1177/0020294018801382.
28. Machado MA. Eddy currents probe design for NDT applications: a review. Sensors. 2024;24(17):5819. doi:10.3390/s24175819.
29. Augustyniak M, Usarek Z. Finite element method applied in electromagnetic NDTE: a review. J Nondestruct Eval. 2016;35(3):39. doi:10.1007/s10921-016-0356-6.
30. Dermani MK, Trajin B. System level modeling of electromagnetic couplings for the energy efficiency of static converters. In: Proceedings of the 2024 IEEE International Conference on Industrial Technology (ICIT); 2024 Mar 25–27; Bristol, UK. p. 1–6. doi:10.1109/ICIT58233.2024.10540973.
31. Soto-Molina I, Ausejo Prieto M, Arce Ruiz RM. Enhancing noise assessment accuracy: FEM as an advanced complement to CNOSSOS-EU. Noise Vib Worldw. 2025;56(3):164–85. doi:10.1177/09574565251319271.
32. Smith BF. Domain decomposition methods for partial differential equations. In: Parallel numerical algorithms. Dordrecht, The Netherlands: Springer; 1997. p. 225–43. doi:10.1007/978-94-011-5412-3_8.
33. Dolean V, Jolivet P, Nataf F. An introduction to domain decomposition methods: algorithms, theory, and parallel implementation. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics; 2015. doi:10.1137/1.9781611974065.
34. Quarteroni A, Valli A. Domain decomposition methods for partial differential equations. Oxford, UK: Clarendon Press; 1999.
35. Centofanti E, Scacchi S. A comparison of Algebraic Multigrid Bidoma in solvers on hybrid CPU-GPU architectures. Comput Meth Appl Mech Eng. 2024;423:116875. doi:10.1016/j.cma.2024.116875.
36. Bell N, Olson LN, Schroder J. PyAMG: algebraic multigrid solvers in Python. J Open Source Softw. 2022;7(72):4142. doi:10.21105/joss.04142.
37. Stüben K. Algebraic multigrid (AMG): experiences and comparisons. Appl Math Comput. 1983;13(3–4):419–51. doi:10.1016/0096-3003(83)90023-1.
38. Phillips TRF, Heaney CE, Smith PN, Pain CC. An autoencoder-based reduced-order model for eigenvalue problems with application to neutron diffusion. Int J Numer Meth Eng. 2021;122(15):3780–811. doi:10.1002/nme.6681.
39. Khoo Y, Lu J, Ying L. Solving parametric PDE problems with artificial neural networks. Eur J Appl Math. 2021;32(3):421–35. doi:10.1017/s0956792520000182.
40. Qin C, Zhao N, Wang X, Li H. An *hp*-adaptive finite-element approach for 3D controlled-source electromagnetic forward modeling. Geophysics. 2025;90(3):WA87–101. doi:10.1190/geo2024-0240.1.
41. Grayver AV, Kolev TV. Large-scale 3D geoelectromagnetic modeling using parallel adaptive high-order finite element method. Geophysics. 2015;80(6):E277–91. doi:10.1190/geo2015-0013.1.
42. Zeng P, Chu G. Residual-aware health prediction of power transformers via spatiotemporal graph neural networks. PLoS One. 2025;20(11):e0332381. doi:10.1371/journal.pone.0332381.
43. Patro A, Tabiei A. A predictive material property scaling approach for eliminating mesh size dependence in numerical simulations for debris impact tests of titanium alloy (TI-6Al-4V) engine casing. J Dyn Behav Mater. 2025;13:1–22. doi:10.1007/s40870-025-00500-x.

44. Zorzetto M, Torchio R, Lucchini F, Di Barba P, Mognaschi ME, Forzan M, et al. Machine learning-based reduced order modeling of nonlinear and multiphysics magnetic devices. IEEE Trans Magn. 2025;PP(99):1. doi:10.1109/TMAG.2025.3626055.
45. Fatima B, Bachir H, Samir B, Karim R, IbnKhaldoun L. Detection and classification of surface cracks using deep learning based autoencoders in eddy current testing. IEEJ Trans Electr Electron Eng. 2025;20(5):676–87. doi:10.1002/tee.24243.
46. Park W, Jo M, Kim M, Lee W. Boundary conditions comparison for electromagnetic simulation using the finite element method with CUDA computing. J Electr Eng Technol. 2024;19(8):5211–20. doi:10.1007/s42835-024-01887-8.
47. Abdelfattah A, Beams N, Carson R, Ghysels P, Kolev T, Stitt T, et al. MAGMA: enabling exascale performance with accelerated BLAS and LAPACK for diverse GPU architectures. Int J High Perform Comput Appl. 2024;38(5):468–90. doi:10.1177/10943420241261960.
48. Betteridge J, Farrell PE, Hochsteger M, Lackner C, Schöberl J, Zampini S, et al. ngsPETSc: a coupling between NETGEN/NGSolve and PETSc. J Open Source Softw. 2024;9(104):7359. doi:10.21105/joss.07359.
49. Raissi M, Perdikaris P, Karniadakis GE. Physics informed deep learning (part I): data-driven solutions of nonlinear partial differential equations. arXiv:1711.10561. 2017.
50. Abdelraouf OAM, Ahmed AMA, Eldele E, Omar AA. Physics-informed neural networks in electromagnetic and nanophotonic design. arXiv:2505.03354. 2025.
51. Hauptmann A, Lucka F, Betcke M, Huynh N, Adler J, Cox B, et al. Model-based learning for accelerated, limited-view 3-D photoacoustic tomography. IEEE Trans Med Imaging. 2018;37(6):1382–93. doi:10.1109/TMI.2018.2820382.
52. Park GH, Krohne K, Bai P, Li EP. Applications of meshfree methods in electromagnetics. In: Proceedings of the IET 7th International Conference on Computation in Electromagnetics (CEM 2008); 2008 Apr 7–10; Brighton, UK. p. 1–2. doi:10.1049/cp:20080204.
53. Li H, Yu M, Jolivet P, Alexandersen J, Kondoh T, Hu T, et al. Reaction-diffusion equation driven topology optimization of high-resolution and feature-rich structures using unstructured meshes. Adv Eng Softw. 2023;180:103457. doi:10.1016/j.advengsoft.2023.103457.
54. Dervovic D, Herbster M, Mountney P, Severini S, Usher N, Wossnig L. Quantum linear systems algorithms: a primer. arXiv:1802.08227. 2018.
55. Pezzè L, Smerzi A. Quantum phase estimation algorithm with Gaussian spin states. PRX Quantum. 2021;2(4):040301. doi:10.1103/prxquantum.2.040301.
56. Gay SE, Ehsani M. Analysis and experimental testing of a permanent magnet eddy-current brake. In: Proceedings of the 2005 IEEE Vehicle Power and Propulsion Conference; 2005 Sep 7; Chicago, IL, USA. p. 1–10. doi:10.1109/VPPC.2005.1554643.
57. Han HS, Kim DS. Magnetic levitation. In: Springer tracts on transportation and traffic. Berlin/Heidelberg, Germany: Springer; 2016.
58. Park J, Pillai N, Wereley NM, Flatau AB. Repulsive magnetic levitation-based electromagnetic energy harvesting of a low-frequency ocean wave. AIP Adv. 2024;14(2):025105. doi:10.1063/9.0000826.
59. Wu M, Yang X, Cui H, Chen W, Wang L, Zhu L, et al. Modeling of litz-wire DD coil with ferrite core for wireless power transfer system. IEEE Trans Power Electron. 2023;38(5):6653–69. doi:10.1109/TPEL.2022.3222228.
60. Zhang X, Zhao Y, Ho SL, Fu WN. Analysis of wireless power transfer system based on 3-D finite-element method including displacement current. IEEE Trans Magn. 2012;48(11):3692–5. doi:10.1109/TMAG.2012.2196263.

61. Tilly J, Chen H, Cao S, Picozzi D, Setia K, Li Y, et al. The Variational Quantum Eigensolver: a review of methods and best practices. Phys Rep. 2022;986:1–128. doi:10.1016/j.physrep.2022.08.003.
62. Zhang Z. Soft-body simulation with CUDA based on mass-spring model and verlet integration scheme. In: Proceedings of the ASME 2020 International Mechanical Engineering Congress and Exposition; 2020 Nov 16–19; Online. p. 1–9. doi:10.1115/IMECE2020-23221.