



Research Proposal

On the Efficacy of Solving the Poisson Equation through a QPINN

A Comparative Study of Quantum and Classical Models as Numerical Solvers

Vyaas Raj

Supervisor: Dr. Mark Hannum

Submitted in partial fulfillment of the research requirements for an Advanced TJHSST Diploma

Contents

1	Introduction	3
1.1	Intellectual Merit	3
1.2	Broader Impact	3
2	Background and Prior Work	3
3	Research Goals	4
	Goal 1: Establish Exact and Classical Baselines	5
	Goal 2: Evaluate a Hybrid Quantum PINN	5
	Goal 3: Implement and Compare an HHL Poisson Solver	5
4	Proposed Methods and Research Design	6
5	Expected Results and Data Analysis	7
6	Feasibility and Resources	8
7	Safety and Risk Considerations	8
8	Potential Pitfalls and Alternative Strategies	8
9	Research Timeline	9
10	Group Responsibilities	10
	Annotated Bibliography	11

1 Introduction

Physics-informed neural networks (PINNs) solve differential equations by training a neural network while penalizing violations of the governing equation and boundary conditions in the loss function [1]. Quantum physics-informed neural networks (QPINNs) extend this idea by embedding a parameterized quantum circuit within part of the classical network [2, 3].

This project will first test how three approaches solve the same one-dimensional Poisson problem on $x \in [-1, 1]$: a classical numerical method, a classical “vanilla” PINN, and a hybrid QPINN containing a quantum layer implemented with the Python library PennyLane. A fourth implementation will discretize the equation and apply the Harrow–Hassidim–Lloyd (HHL) quantum linear-systems algorithm as a distinct quantum baseline [4]. The prescribed profile $u_0(x) = \sin(\pi x)$ will be treated as a manufactured exact solution, since the time-independent Poisson equation has boundary conditions rather than an initial condition. After the 1D implementations are validated, the numerical, PINN, and QPINN pipelines will be extended to a 2D Poisson equation on $[-1, 1]^2$; HHL will also be tested in 2D if the required circuit is worth it. We will measure relative L^2 solution error against the exact solution, physics residual, convergence, training time, circuit resources, and variation across repeated runs. The study will allow the quantum methods to improve, match, or underperform the classical methods.

1.1 Intellectual Merit

The project will isolate the effect of a quantum layer through controlled benchmarking rather than evaluating whether a hybrid model can produce a reasonable solution. Comparing the QPINN with a matched classical PINN, an exact solution, a conventional discretization, and HHL may clarify whether observed differences arise from representation, parameter count, optimization dynamics, discretization, quantum noise, or computational overhead.

1.2 Broader Impact

Clear negative or neutral results are useful in quantum machine learning because they help identify where quantum components should be used and whether their use is justified. The benchmark, code, and analysis workflow developed here could also be reused by future QLab projects evaluating hybrid scientific-machine-learning models or quantum algorithms for computational fluid dynamics (CFD) and partial differential equations more generally.

2 Background and Prior Work

The Poisson equation relates the curvature of an some unknown field to a known source and appears in electrostatics, gravitation, diffusion, and pressure-correction methods for incompressible fluid flow. This project will study the one-dimensional Dirichlet problem:

$$-\frac{d^2u}{dx^2} = f(x), \quad x \in [-1, 1], \quad u(-1) = u(1) = 0. \quad (1)$$

To keep the specified condition similar (using $u_0(x) = \sin(\pi x)$ as our condition) in a time-independent setting, we use the method of manufactured solutions (a powerful technique ized to verify computer programs that solve for some PDE): we shall define $u_{\text{exact}}(x) = \sin(\pi x)$ and therefore $f(x) = \pi^2 \sin(\pi x)$. Note, this is done because according to the Poisson Equation, we must take the negative of the double derivative. The exact solution satisfies both endpoint values and

gives us our accuracy reference. A centered finite-difference discretization converts the differential equation into a sparse symmetric positive-definite linear system $A\mathbf{u} = \mathbf{b}$. This format will help us use the HHL algorithm – but more on that later.

After validating the one-dimensional test case, the project will consider the two-dimensional extension

$$-\nabla^2 u(x, y) = f(x, y), \quad (x, y) \in [-1, 1]^2, \quad (2)$$

with $u = 0$ on the square boundary. The solution $u_{\text{exact}}(x, y) = \sin(\pi x)\sin(\pi y)$ gives $f(x, y) = 2\pi^2 \sin(\pi x)\sin(\pi y)$. This preserves the specified sinusoidal case that we previously used while also adding two-coordinate encoding, a four-edge boundary loss, and a much larger finite-difference system. Classical PINN software supports multidimensional domains and automatic differentiation (which is useful for our methodology) [5], and prior QPINN research has directly studied manufactured 2D Poisson solutions using PennyLane circuits [2].

A classical PINN represents $u(x)$ with a differentiable neural network and minimizes a loss combining the interior residual $-u''_{\theta}(x) - f(x)$ with errors at $x = -1$ and $x = 1$ [1]. Automatic differentiation supplies the second derivative. Although labeled solution data are not required for training, PINNs can exhibit imbalanced loss gradients and difficult optimization landscapes, so learning rate, component losses, and convergence behavior should be measured rather than treated as incidental implementation details [6, 7].

In the proposed hybrid model, a classical encoder maps x to rotation angles, a variational circuit applies angle encoding, trainable rotations, and entangling gates (it is to be noted that it occurs in a Bloch Sphere), and measured expectation values pass through a classical decoder. Gradients must support both optimization with respect to circuit parameters and spatial differentiation with respect to x . Prior work has demonstrated differentiable quantum circuits for differential equations and quantum or hybrid PINNs for one- and two-dimensional PDEs, which also includes Poisson problems [2, 3]. Reported parameter efficiency does not establish a general advantage, and circuit evaluation can be substantially slower than a comparable classical network [2]. This shows that – at times – a hybrid model is not the necessarily best choice, depending on whatever a scenario requires.

HHL takes a different route. After discretization, it prepares a quantum state proportional to the solution of a suitable linear system rather than learning a continuous coordinate-to-solution map [4]. Poisson-specific quantum algorithms exploit the structured sparse matrix produced by finite differences [8, 9], and related decompositions have been developed explicitly for d -dimensional Dirichlet Poisson systems [10]. However, state preparation, conditioning, circuit depth, sampling, and extraction of the full solution can remove an apparent theoretical speedup. Real-device studies of small Poisson systems therefore provide an important caution for this comparison [11].

Project Need / Research Gap. Prior studies show that PINNs, QPINNs, and quantum linear solvers can address Poisson-type equations, but they are often evaluated under different assumptions and metrics. This project will place an exact solution, a conventional discretization, a classical PINN, a hybrid QPINN, and a small HHL implementation on transparent 1D and 2D benchmarks. The staged design will show not only whether a method solves the simple problem, but also how its accuracy and resource requirements change when moving from a line to a square domain.

3 Research Goals

Research goals describe what will be determined, not what the authors hope the quantum model will do. All model comparisons will therefore report improvements, no meaningful differences, and

degradations with equal emphasis.

Goal 1— Establish Exact and Classical Baselines

Question / Objective: Determine the discretization error and training behavior of conventional numerical and PINN solvers for the manufactured Poisson problem.

Approach: We shall evaluate $u_{\text{exact}}(x) = \sin(\pi x)$ on a dense grid, solve the centered finite-difference system at several grid resolutions, and then train a classical fully connected PINN across a small learning-rate sweep. After validating the 1D code, repeat the workflow for $u_{\text{exact}}(x, y) = \sin(\pi x) \sin(\pi y)$ on $[-1, 1]^2$. We shall then repeat stochastic training with at least five random seeds while holding collocation points, stopping rules, and evaluation points fixed within each dimension.

Measurement / Output: 1D and 2D reference solutions and distributions for relative L^2 error, maximum absolute error, boundary error, PDE residual, epochs to threshold, wall time, and seed-to-seed variation.

Success Criterion: The baseline is complete when the finite-difference solutions show the expected grid-refinement trend and the 1D and 2D classical PINN results, loss histories, configurations, and random seeds are reproducible.

Goal 2— Evaluate a Hybrid Quantum PINN

Question / Objective: Determine what changes when a classical hidden layer is replaced by a small parameterized quantum circuit.

Approach: Construct a classical encoder–quantum circuit–classical decoder architecture based on the attached framework. Compare it with the classical PINN and, if practical, a parameter-matched classical bottleneck. Validate the circuit on the 1D problem, then encode both x and y for the 2D problem. Test a limited matrix of qubit counts, circuit depths, and learning rates while retaining identical collocation and evaluation points within each benchmark.

Measurement / Output: Matched 1D and 2D comparisons of relative L^2 error, maximum and boundary error, PDE residual, convergence curves, wall-clock time, parameter count, circuit evaluations, and stability across seeds. Quantum expectation values and gradient norms will be logged when practical.

Success Criterion: The goal is complete when the hybrid and classical PINNs have been evaluated under the same protocol in 1D and at least one feasible 2D configuration has been evaluated. A null or negative quantum result satisfies this goal.

Goal 3— Implement and Compare an HHL Poisson Solver

Question / Objective: Determine what solution information a small HHL implementation can recover from the discretized Poisson equation and at what resource cost.

Approach: Construct the 1D finite-difference matrix A and vector $\mathbf{b}$ at power-of-two dimensions suitable for amplitude encoding. Run HHL first on a noiseless simulator and then, if circuit depth and access permit, on available quantum hardware. After validation, construct the 2D Kronecker-sum discretization and test the smallest feasible instance in simulation, extending to hardware only if circuit resources permit. Compare the reconstructed or sampled solution observables with the exact and classical discrete solutions.

Measurement / Output: State fidelity or solution-vector error where reconstruction is possible, selected observable errors, qubit count, circuit depth, shot count, success probability, wall time, and sensitivity to device noise.

Success Criterion: Produce a reproducible small-instance HHL demonstration and an honest accounting of which costs are included. If hardware noise prevents an accurate solution but the simulator succeeds, documenting that separation satisfies the goal.

4 Proposed Methods and Research Design

The benchmark will be:

$$-u''(x) = \pi^2 \sin(\pi x), \quad x \in [-1, 1], \quad u(-1) = u(1) = 0, \quad (3)$$

with exact solution $u_{\text{exact}}(x) = \sin(\pi x)$. Although the previous equation was described as an initial condition, it is used as a manufactured exact solution because this Poisson problem has no time coordinate. A dense fixed grid will be used only for evaluation. Centered finite differences on the interior points will provide the conventional numerical solution and the matrix system used by HHL.

The second stage will use the square domain $\Omega = [-1, 1]^2$ and

$$-(u_{xx} + u_{yy}) = 2\pi^2 \sin(\pi x) \sin(\pi y), \quad u|_{\partial\Omega} = 0, \quad (4)$$

whose exact solution is $u_{\text{exact}}(x, y) = \sin(\pi x) \sin(\pi y)$. A five-point finite-difference stencil will provide the 2D numerical baseline. Interior collocation points will be sampled over the square, while boundary points will be sampled on all four edges. This manufactured benchmark follows the same general 2D Poisson strategy used in prior QPINN work [2] while retaining the sinusoidal form specified for this project.

The classical PINN will be a fully connected network with hyperbolic-tangent activations. The hybrid QPINN will follow the following architecture: a classical encoder scales the coordinate or coordinate pair into an angle range, a PennyLane circuit applies angle encoding, trainable rotations, and entanglers, and measured Pauli-Z expectation values enter a classical decoder. In 2D, the loss replaces u''_θ with $u_{\theta,xx} + u_{\theta,yy}$ and includes all four boundary edges. The first implementation will use a small register and shallow ansatz; qubit count and depth will increase only if nested spatial and parameter differentiation remains tractable.

Adam will be used for initial PINN and QPINN training, with a modest learning-rate sweep and at least five seeds for retained configurations. Matched comparisons will use the same collocation points, loss weights, epoch budget, stopping thresholds, and evaluation grid. If useful, collocation points may be periodically resampled, but that policy will be shared by both models. A parameter-matched classical bottleneck will be included if its construction does not obscure the main comparison.

For HHL, the finite-difference system will be normalized and embedded at a power-of-two dimension. The 2D matrix will be constructed as a Kronecker sum of 1D discrete Laplacians, so even a modest grid produces many more unknowns than the 1D case. Small instances will be tested on a noiseless simulator before execution on available hardware. Because HHL prepares a state proportional to $A^{-1}\mathbf{b}$ rather than directly printing every component, the study will distinguish state fidelity and selected observables from the additional tomography or sampling needed to reconstruct a full vector [4, 10, 11].

Data analysis will use relative L^2 error, maximum absolute error, boundary error, and residual error on a common grid. Two-dimensional results will additionally include solution and error heat maps plus centerline slices at $x = 0$ and $y = 0$. For learned models the analysis will report total and component losses, epochs to threshold, wall time, parameter count, gradient behavior, and failures. Means, medians, standard deviations, and individual-run plots will summarize repeated seeds. For HHL, qubits, depth, shots, success probability, and hardware noise will be reported separately so unlike output models are not collapsed into a misleading single ranking.

Methods may change. These procedures describe the current experimental design. If nested differentiation through the quantum layer is impractical, HHL circuits exceed available hardware, or a hyperparameter range is inappropriate, the relevant experiment will be reduced to smaller instances or simulation. Every revision and its effect on comparability will be documented.

5 Expected Results and Data Analysis

The exact and finite-difference solutions are expected to provide clean validation targets in both dimensions. For centered second-order discretizations, error should decrease predictably as the grid is refined until floating-point effects dominate. The classical PINN should recover the smooth sinusoidal solutions, although the 2D problem requires broader collocation coverage and enforcement on four boundary edges. Accuracy and convergence may still depend on learning rate and on the balance between residual and boundary losses [6, 7].

For the QPINN, no direction of effect is assumed. It may achieve lower error or a similar error with fewer trainable parameters, as reported for some prior QPINN tests, including manufactured 2D Poisson equations [2].

Small noiseless HHL instances are expected to recover a state close to the normalized discrete solution. The 2D discretization is expected to reach resource limits at a much smaller number of grid points per axis because the number of unknowns grows with the product of the grid dimensions. On real hardware, greater error is expected from circuit depth, finite sampling, and device noise, consistent with published Poisson benchmarks on current quantum devices [11].

For each learned configuration, metrics will be summarized across random seeds using the mean, median, standard deviation, and full range where useful. Accuracy-versus-epoch and accuracy-versus-wall-time curves will distinguish optimization behavior from raw compute cost. Grid-refinement plots will be used for the classical discretization; fidelity or observable error versus shots and noise will be used for HHL. Comparisons will emphasize effect sizes and uncertainty rather than one best run.

The most informative result may be a tradeoff rather than a simple ranking. A QPINN could reduce parameter count while increasing wall time, or HHL could represent the normalized solution compactly while making full readout expensive. These effects will be reported separately.

6 Feasibility and Resources

The project requires no custom laboratory apparatus. Both manufactured Poisson problems have closed-form solutions and inexpensive classical references, so no external labeled dataset is required. The necessary “data” are generated interior collocation points, boundary points, source values, and fixed 1D or 2D evaluation grids. Development requires Python, an automatic-differentiation framework, PennyLane or an equivalent quantum-machine-learning interface, simulators, and access to the quantum computers already available to the group.

The classical numerical solutions and PINNs are feasible on a regular NVIDIA GPU, although the 2D parameter sweeps will require more collocation points and training time. The main constraints are QPINN differentiation cost and HHL circuit depth, shots, state preparation, and read-out; these costs grow sharply when moving to 2D. Access to a CUDA-Q GPU will be greatly helpful in this process as it shall allow us to accurately compute our QPINN algorithm(s).

7 Safety and Risk Considerations

This project does not involve lasers, high voltage, vacuum systems, chemicals, or other significant laboratory hazards. The primary risks are computational and operational rather than physical. Long GPU jobs will be executed only on approved systems with normal thermal management and resource limits; we, as students, will not modify workstation cooling, power hardware, or system-level GPU settings.

8 Potential Pitfalls and Alternative Strategies

Challenge 1

The classical PINN may train unreliably or satisfy the interior residual while missing the boundary conditions, making a QPINN comparison ambiguous.

Solution: Validate the exact and finite-difference implementations first, monitor residual and boundary losses separately, and complete a small learning-rate and loss-weight study before freezing the classical protocol used for hybrid comparisons.

Challenge 2

Second spatial derivatives through the quantum layer may add so much simulation, sampling, and differentiation cost that the planned QPINN sweep becomes impractical.

Solution: Begin with a small register and shallow circuit, profile one complete training run before expanding the sweep, and prioritize repeated matched comparisons over many architectures. If hardware training is impractical, train on a differentiable simulator and reserve hardware for forward-pass validation of selected circuits.

Challenge 3

A QPINN may appear better only because it has a different parameter count, because the sinusoidal target aligns unusually well with angle encoding, or because one initialization performs unusually well.

Solution: Use identical points and stopping rules, include a parameter-matched classical bottleneck if practical, repeat retained configurations across multiple seeds, and limit conclusions to this benchmark. Report distributions rather than only the best run.

Challenge 4

HHL may require circuits deeper than available hardware can execute accurately, and reconstructing the full solution vector may require many measurements.

Solution: Start with the smallest power-of-two discretizations on a noiseless simulator, record circuit resources and state fidelity, and then attempt only hardware sizes justified by those measurements. If full tomography is impractical, compare selected observables while clearly distinguishing them from full-vector recovery.

Challenge 5

The selected quantum feature encoding or ansatz may produce nearly constant features, vanishing gradients, or unstable gradients.

Solution: Record expectation values and gradient norms during training. If the layer is demonstrably nontrainable, test one pre-specified alternative encoding or entangling pattern rather than repeatedly redesigning the circuit until a favorable result appears.

Challenge 6

The 1D implementations may succeed while the 2D versions become too expensive or inaccurate because collocation coverage, derivative evaluations, matrix dimension, and circuit resources all increase.

Solution: We treat 1D validation as the required baseline and 2D as a staged extension. We shall then begin with coarse 2D grids and one retained architecture, refine only after correctness is established, and report the largest feasible 2D problem rather than silently replacing repeated comparisons with one favorable run.

9 Research Timeline

Milestone 1: End of Quarter 1: Mathematical Benchmark Established

Task — What will you do? Derive the 1D and 2D manufactured Poisson sources, implement the exact and finite-difference solutions, define common grids and metrics, and verify the sign and boundary conventions.

Objective — What measurable result should this work produce? Produce tested 1D and 2D reference solutions and grid-refinement results with logged relative L^2 , maximum, boundary, and residual errors.

Goal — What larger research capability or milestone does this establish? Establish a trustworthy benchmark against which every learned and quantum result can be evaluated.

Milestone 2: End of 3rd Quarter: Classical PINN Baseline Complete

Task — What will you do? Implement the 1D classical PINN, validate automatic second derivatives, tune a limited learning-rate and boundary-loss grid, and then extend the retained architecture and loss to the 2D square domain.

Objective — What measurable result should this work produce? Complete at least five repeated 1D runs for each retained setting and a matched repeated 2D baseline with logged errors, component losses, convergence curves, parameter count, and wall time.

Goal — What larger research capability or milestone does this establish? Freeze a reproducible classical PINN protocol before judging the quantum layer.

Milestone 3: End of 3rd Quarter: Quantum Implementations Complete

Task — What will you do? Integrate the variational quantum layer into the PINN, complete matched 1D and feasible 2D simulator runs, implement HHL for small 1D and 2D finite-difference systems, and attempt selected runs on available quantum hardware.

Objective — What measurable result should this work produce? Produce repeated QPINN comparisons in both dimensions plus feasible HHL fidelity or observable results with qubit, depth, shot, noise, and timing records.

Goal — What larger research capability or milestone does this establish? Determine what the QPINN and HHL approaches can accurately return as the Poisson benchmark expands from one to two dimensions and at what cost.

Milestone 4: Mid-3rd Quarter: Final Analysis and Research Product

Task — What will you do? Complete statistical comparisons, regenerate figures from archived runs, document deviations from the proposed methods, and write the final paper and presentation.

Objective — What measurable result should this work produce? Produce a reproducible package containing source code, configurations, seeds, backend records, benchmark tables, final figures, and a conclusion addressing each research goal.

Goal — What larger research capability or milestone does this establish? Deliver a defensible comparison of exact, conventional numerical, PINN, QPINN, and HHL approaches, including null or negative quantum results.

10 Group Responsibilities

Vyaas Raj

I shall undertake primary responsibility for writing and testing the 1D and feasible 2D HHL implementations and for building the hybrid QPINN by adding and integrating the variational quantum layer. I will document circuit architecture, coordinate encoding, differentiation method, qubit and depth requirements, backend settings, shot counts, and quantum-hardware results.

Mohith Thathineni

He will be primarily be responsible for coding and validating the 1D and 2D classical PINNs, including automatic differentiation of the Poisson residual, boundary-condition losses, optimizer

settings, learning-rate tests, and repeated-seed training runs.

Arnabh Rao

He will be primarily be responsible for coding the 1D and 2D conventional classical numerical solutions and exact manufactured solutions. Arnabh will also implement the finite-difference discretizations, grid-refinement checks, common evaluation grids, and reference error calculations.

All three of us will [together] confirm the mathematical conventions, define shared input and output formats, review failed or interesting runs, compare results using the same metrics, interpret the findings, and contribute to the final paper and presentation. Each of us is expected to understand the exact, numerical, PINN, QPINN, and HHL approaches well enough to evaluate the complete comparison.

Annotated Bibliography

- [1] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”, *Journal of Computational Physics* **378**, 686–707 (2019).

What did this work do? Introduced the modern PINN framework, in which governing differential equations and boundary or initial conditions are incorporated into a neural-network loss through automatic differentiation.

Why is it relevant? It provides the foundational classical method used by both the PINN and hybrid QPINN in this proposal.

How will we use it? We will use its physics-informed residual construction to train a coordinate-to-solution network without labeled interior solution data.

- [2] C. Trahan, M. Loveland, and S. Dent, “Quantum physics-informed neural networks”, *Entropy* **26**, 649 (2024).

What did this work do? Compared classical, quantum, and hybrid physics-informed neural networks on steady and transient one- and two-dimensional differential equations, including Poisson examples.

Why is it relevant? It is the closest methodological precedent for the proposed PennyLane QPINN and reports both parameter-efficiency opportunities and substantial timing overhead.

How will we use it? We will adapt its angle-encoded variational-layer concept and compare accuracy, parameter count, initialization sensitivity, and wall time rather than accuracy alone.

- [3] O. Kyriienko, A. E. Paine, and V. E. Elfving, “Solving nonlinear differential equations with differentiable quantum circuits”, *Physical Review A* **103**, 052416 (2021).

What did this work do? Represented trial functions with parameterized quantum circuits and used analytical differentiation to train them against differential equations and boundary conditions.

Why is it relevant? The proposed QPINN likewise requires derivatives of a quantum-circuit output with respect to spatial inputs as well as gradients with respect to trainable parameters.

How will we use it? We will use its differentiable-circuit perspective when designing and diagnosing the quantum layer and its spatial derivatives.

- [4] A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations”, *Physical Review Letters* **103**, 150502 (2009).

What did this work do? Introduced the HHL quantum linear-systems algorithm, which prepares a quantum state proportional to the solution of a sparse, well-conditioned linear system under specified access assumptions.

Why is it relevant? Finite-difference discretization converts the Poisson equation into the type of linear system that motivates the project’s HHL baseline.

- How will we use it?** We will implement HHL only for small discretizations and will report state preparation, conditioning, success probability, and readout limitations alongside accuracy.
- [5] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, “DeepXDE: a deep learning library for solving differential equations”, *SIAM Review* **63**, 208–228 (2021).
What did this work do? Presented a software library for physics-informed and data-driven solution of ordinary and partial differential equations across multidimensional geometries and boundary conditions.
Why is it relevant? The proposed extension requires a classical PINN to evaluate a two-dimensional Laplacian and impose boundary losses on all four edges of a square domain.
How will we use it? We will use its multidimensional PINN formulation as a reference for structuring collocation points, boundary conditions, automatic differentiation, and evaluation in the 2D implementation.
- [6] S. Wang, Y. Teng, and P. Perdikaris, “Understanding and mitigating gradient flow pathologies in physics-informed neural networks”, *SIAM Journal on Scientific Computing* **43**, A3055–A3081 (2021).
What did this work do? Analyzed imbalanced gradient flow among PINN loss terms and proposed an annealing method based on gradient statistics.
Why is it relevant? The residual and boundary losses in the proposed Poisson PINN may converge at different rates even though the exact solution is smooth.
How will we use it? We will log component losses and gradients and will use a documented weighting or learning-rate adjustment if the baseline shows this pathology.
- [7] A. S. Krishnapriyan, A. Gholami, S. Zhe, R. M. Kirby, and M. W. Mahoney, “Characterizing possible failure modes in physics-informed neural networks”, in *Advances in neural information processing systems*, Vol. 34 (2021), pp. 26548–26560.
What did this work do? Demonstrated that optimization can prevent PINNs from solving even well-defined differential-equation problems and examined strategies for mitigating those failures.
Why is it relevant? A failed QPINN comparison is uninterpretable unless the classical PINN and its training protocol have first been validated.
How will we use it? We will establish the exact and classical baselines first, track stalled runs, and separate optimization failure from representational error.
- [8] Y. Cao, A. Papageorgiou, I. Petras, J. F. Traub, and S. Kais, “Quantum algorithm and circuit design solving the poisson equation”, *New Journal of Physics* **15**, 013021 (2013).
What did this work do? Developed a quantum algorithm and explicit circuit elements for solving a discretized Poisson equation using a quantum linear-systems approach.
Why is it relevant? It connects the general HHL framework directly to the structured finite-difference matrix used in this project.
How will we use it? We will use it to guide the mapping from the Poisson discretization to a small quantum linear-system circuit and to identify resource assumptions.
- [9] H.-L. Liu, Y.-S. Wu, L.-C. Wan, S.-J. Pan, S.-J. Qin, F. Gao, and Q.-Y. Wen, “Variational quantum algorithm for the poisson equation”, *Physical Review A* **104**, 022418 (2021).
What did this work do? Proposed a near-term variational solver for finite-difference Poisson systems using a compact decomposition of the coefficient matrix.
Why is it relevant? It provides a hardware-oriented alternative to fault-tolerant HHL and shows how the Poisson matrix structure can reduce measurement terms.
How will we use it? If HHL depth is prohibitive, this method provides a documented alternative strategy while keeping the same discretized benchmark.

- [10] H.-M. Li, Z.-X. Wang, and S.-M. Fei, “Variational quantum algorithms for poisson equations based on the decomposition of sparse hamiltonians”, *Physical Review A* **108**, 032418 (2023).
What did this work do? Developed sparse-Hamiltonian decompositions and quantum circuits for one-dimensional Poisson systems with several boundary conditions and for d -dimensional Dirichlet Poisson systems.
Why is it relevant? It directly addresses how the structured linear system changes when the proposed quantum Poisson study is extended from one dimension to two.
How will we use it? We will use its multidimensional resource analysis to guide the smallest feasible 2D quantum linear-system experiment and to explain any simulator or hardware limits.
- [11] M. Ali and M. Kabel, “Performance study of variational quantum algorithms for solving the poisson equation on a quantum computer”, *Physical Review Applied* **20**, 014054 (2023).
What did this work do? Benchmarked variational Poisson solvers on real quantum devices and found that results promising on noiseless simulators degraded substantially on current hardware.
Why is it relevant? It demonstrates why simulator accuracy, hardware accuracy, amplitude encoding, sampling, and full solution readout must be distinguished in a quantum Poisson study.
How will we use it? We will report simulator and hardware outcomes separately and include circuit resources, shots, noise, and the cost of extracting solution information.