

Are Physics-Informed Neural Networks the Future of Financial Modelling?

Reuben Allen

10824401

The University of Manchester

B.Sc. Dissertation

March 2025

Abstract

Physics-Informed Neural Networks (PINNs) are neural networks that embed partial differential equations (PDEs) directly into their architecture, enabling solutions that are consistent with underlying physical laws. This dissertation demonstrates the accuracy of PINNs in solving wave equations in one and three dimensions, including nonlinear and periodic boundary conditions, confirming robust generalisation under data scarcity. These insights are then transferred to finance, where many pricing models reduce to PDEs. Subsequently, PINNs are applied to the Black–Scholes equation and extended to model stochastic volatility using time-dependent parameters. Results show that PINNs, through automatic differentiation and mesh-free formulation, can produce interpretable, accurate solutions while maintaining performance under covariate shift. This work supports PINNs as a scalable and promising framework for modern financial modelling.

Contents

1 Introduction	3
2 Background on Neural Networks	4
2.1 Standard Architectures	4
2.2 Embedding Physics into Neural Networks	5
3 Solving Wave Equations with PINNs	6
3.1 The 1D Wave Equation	6
3.2 Scaling to 3D	7
3.3 Handling Nonlinearity and Boundary Conditions	8
4 Applications to Financial Modelling	9
4.1 Option Pricing with Black–Scholes	9
4.2 Modelling Stochastic Volatility	10
4.3 Learning from Noisy Market Data	11
5 Comparison with Traditional Methods	13
5.1 Experimental Insights	13
5.2 Limitations and Practical Considerations	14
6 Discussion	15
7 Conclusion	17
Appendix A: Wave Equation Experiments	19
Appendix B: Financial Modelling Experiments	21
Appendix C: Quantitative Performance Summary	23
References	24

1 Introduction

Partial differential equations (PDEs) form the mathematical foundation for modelling dynamical systems across physics, engineering, and finance. In quantitative finance, derivative pricing models often reduce to PDEs, notably the Black–Scholes equation for European options [1]. Traditional numerical approaches, including finite difference schemes [2], Monte Carlo simulations [3], and finite element methods—while effective for structured problems—typically require re-discretisation when domain geometry, boundary conditions, or volatility structures change.

Physics-Informed Neural Networks (PINNs) offer a novel methodology by incorporating partial differential equation (PDE) restrictions directly into the loss function of the neural network [4]. Training involves sampling collocation points and penalising any deviation from the underlying physics, whether due to the equation itself or the imposed constraints. Unlike standard neural architectures—such as convolutional neural networks [5] or recurrent variants—PINNs encode the structure of governing equations into their training objective. This acts as a form of inductive bias, improving generalisation and interpretability in low-data regimes [6].

Recent work has applied PINNs to a variety of physical domains, including wave propagation, quantum mechanics, and fluid dynamics [7]. Their mesh-free nature makes them particularly appealing for irregular geometries and evolving boundary conditions. While their use in physics is growing rapidly, applications to financial modelling remain comparatively underdeveloped. This is because financial systems are often noisy and nonlinear, and must respect theoretical constraints such as no-arbitrage and risk-neutral pricing [8].

Standard neural networks, despite success in empirical financial prediction, often suffer from overfitting and lack interpretability in stochastic systems [9]. PINNs provide as a possible link between conventional modeling and data-driven learning, integrating interpretability with adaptability.

This dissertation examines the feasibility of Physics-Informed Neural Networks (PINNs) as adaptable and generalisable solvers for partial differential equations (PDEs) in the field of finance. Primary focal points is their capacity to derive analytical solutions, manage irregular or dynamic boundary circumstances, and sustain performance throughout covariate shift [10]. Structural priors—such as periodic boundary conditions and nonlinear residual terms—are introduced to assess how well PINNs generalise beyond the training domain, while hybrid loss functions are used to combine physical constraints with empirical data. The impact of these design choices is evaluated by tracking model convergence and performance across different architectures, using a combination of visualisations and error metrics. This investigation is carried out through two structured notebooks:

Notebook A assesses canonical partial differential equations in physics, progressively augmenting complexity through nonlinearity and periodicity, while juxtaposing results with analytical benchmarks.

Notebook B expands upon these findings inside financial systems by addressing the traditional Black-Scholes partial differential equation, incorporating stochastic volatility, and evaluating robustness using synthetic noisy data. This tests whether techniques effective in modelling physical PINNs transfer well to financial settings.

A consistent challenge is generalisation outside the training domain. PINNs trained under one regime may underperform when exposed to volatility patterns or boundary behaviours not seen during training. This motivates investigation into adaptive priors and hybrid models for robustness.

The majority of experiments in the literature use the Adam optimiser [11] with 5000 training epochs. Visualisations of convergence and accuracy are added to supplement performance analysis. Although PINNs currently lack the theoretical guarantees offered by classical solvers, they open compelling new avenues for dynamic financial modelling [12], including future extensions via quantum computation [13].

In summary, this dissertation explores whether PINNs can offer a credible and flexible alternative to traditional numerical solvers and standard neural networks—particularly in financial domains characterised by noise, structural change, and limited data.

2 Background on Neural Networks

2.1 Standard Architectures

Artificial neural networks (ANNs) are universal function approximators composed of layers of interconnected units (neurons). Each layer implements a linear transformation followed by a nonlinear activation, mapping an input vector \mathbf{x} to an output y :

$$y = \phi(W_2 \cdot \phi(W_1 \cdot \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2), \quad (1)$$

where W_i and \mathbf{b}_i in this context denote trainable weights and biases, and ϕ is a nonlinear activation function, typically the rectified linear unit (ReLU) or hyperbolic tangent (tanh) [14]. ReLU is favoured for its computational simplicity and ability to alleviate vanishing gradient problems, defined as:

$$\text{ReLU}(z) = \max(0, z). \quad (2)$$

Training these networks involves minimising a loss function using gradient-based optimisation methods such as stochastic gradient descent (SGD) or the Adam optimiser [11]. SGD updates network parameters continually using gradients computed from

mini-batches of data. This provides a balance between convergence speed and memory efficiency [15]. ‘Adam’ improves upon this by adjusting learning rates adaptively for each parameter based on estimates from the first and second moments of the gradients.

ANNs have been shown to be successful across many domains—including computer vision, natural language processing, and fluid dynamics [16]—but their performance hinges on access to large, representative datasets. In sparse or noisy regimes, such as in scientific modelling, they tend to overfit and fail to extrapolate outside the training domain [5]. This shortcoming motivates the development of architectures that can embed physical or even structural knowledge into the training process.

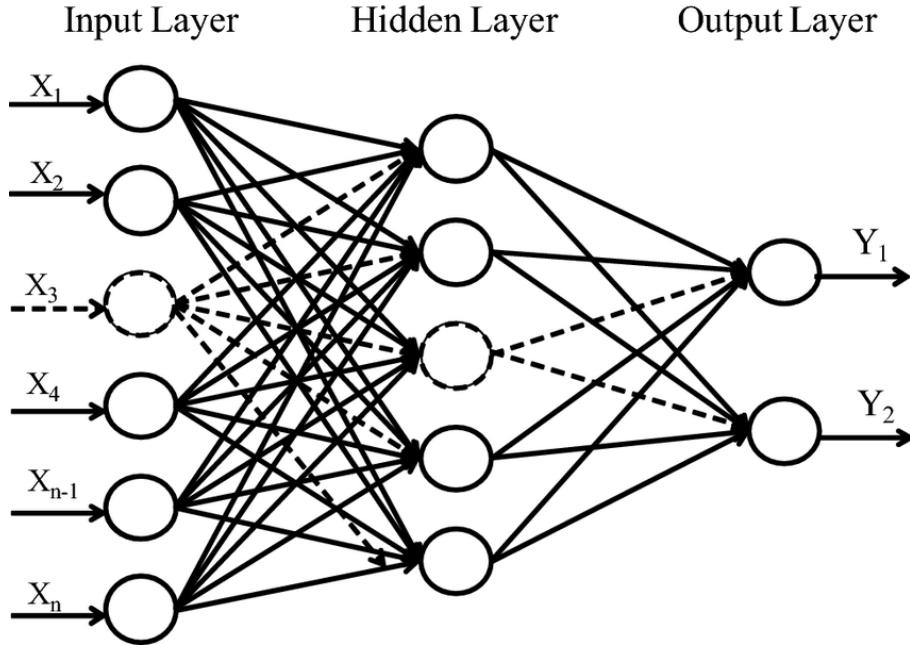


Figure 1: Feedforward ANN architecture applied to hydrodynamic modelling [16].

2.2 Embedding Physics into Neural Networks

Physics-Informed Neural Networks (PINNs) integrate physical laws directly into the training process of neural networks [4]. Unlike conventional models that rely purely on labelled data, PINNs enforce compliance with known governing equations by penalising residuals across randomly sampled collection points within the solution domain. This enables them to solve partial differential equations (PDEs) without requiring explicit mesh generation or traditional discretisation.

The total loss function used to train a PINN of this nature typically combines three main components:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{PDE}} \mathcal{L}_{\text{PDE}} + \lambda_{\text{IC}} \mathcal{L}_{\text{IC}} + \lambda_{\text{BC}} \mathcal{L}_{\text{BC}}, \quad (3)$$

where \mathcal{L}_{PDE} corresponds to the residual of the PDE itself. \mathcal{L}_{IC} enforces initial conditions, and \mathcal{L}_{BC} imposes boundary constraints. The weighting coefficients λ_i provide the key

flexibility needed in tuning how much influence each term has during optimisation [17].

A core enabler of PINNs is automatic differentiation, which allows for the efficient and exact computation of derivatives required by the PDE residual [18]. This differentiates PINNs from numerical solvers that approximate derivatives using finite differences. The result is a mesh-free approach that is well suited to modelling dynamic domains and irregular boundary conditions [19].

Recent extensions have also shown that PINNs can be applied to problems involving stiff PDEs and noisy systems, which are crucial to fields such as quantum mechanics [20]. Such studies illustrate how PINNs can outperform traditional solvers, especially in contexts where the ability to encode important structural properties flexibly is essential.

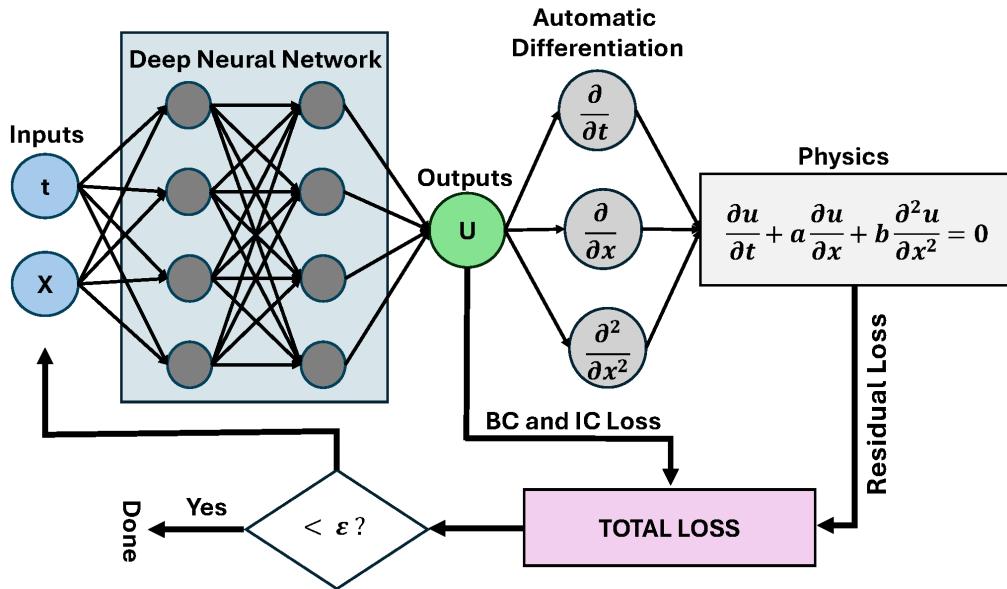


Figure 2: Typical PINN architecture for PDE solving, illustrating how automatic differentiation computes derivatives directly for physics-based loss evaluation [20].

3 Solving Wave Equations with PINNs

3.1 The 1D Wave Equation

The classical wave equation models how waves propagate through various media and remains a cornerstone of modern physics. Including fields like mathematical physics, acoustics and electromagnetism. As a result we wish to create a PINN implementation to model the 1D wave equation, as this will test PINNs viability in solving fundamental physical problems. In one spatial dimension, it is given by:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}, \quad (4)$$

where $u(x, t)$ denotes the wave amplitude and c is the wave propagation speed [21, 22]. Analytical solutions for PDEs of this kind are well established, thus enabling direct validation of our numerical approximations.

To solve this equation using a Physics-Informed Neural Network (PINN), a feedforward neural network $u_\theta(x, t)$ is trained to approximate the solution over the spatiotemporal domain. This strategy, in contrast to standard methods, does not depend on grid discretisation or mesh generation [4]. Instead, the network is optimised to minimise a compound loss function that incorporates the PDE residual, alongside its initial conditions, and Dirichlet boundary conditions. All evaluated at randomly sampled collection points.

Automatic differentiation plays a critical role in PINN training. By computing required spatial and temporal derivatives directly from the network's computational graph [18], it allows seamless integration of physical constraints into the learning process—without numerical differentiation or finite difference approximations.

The trained model successfully recovered the known analytical solution:

$$u(x, t) = \sin(\pi x) \cos(\pi t), \quad (5)$$

demonstrating high predictive accuracy and rapid convergence. The final mean square error (MSE) was below 1.15×10^{-4} , confirming the capacity of the model to effectively approximate the wave solution. These results align with other recent evaluations found with PINNs for linear PDEs [7]. This result further showcases their potential for mesh-free scientific computing.

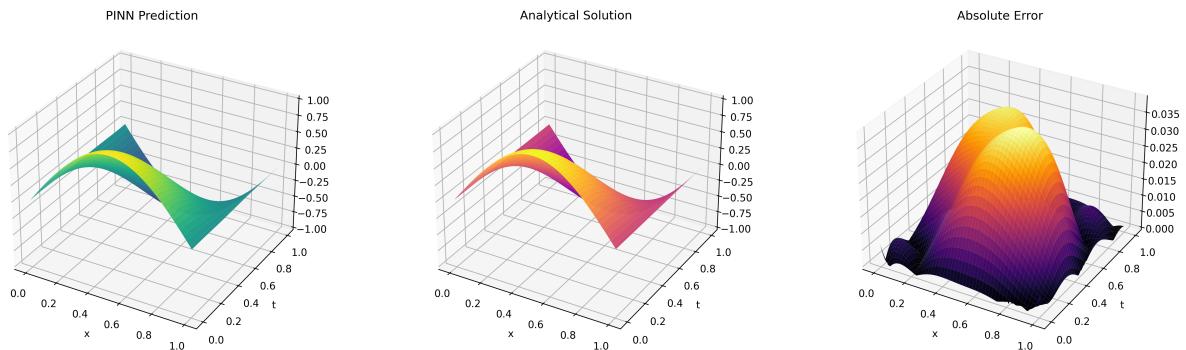


Figure 3: Comparison of PINN prediction (left), analytical solution (middle), and absolute error (right) for the 1D wave equation.

3.2 Scaling to 3D

Extending the PINNs to three spatial dimensions significantly increases both computational complexity and training difficulty. The corresponding wave equation is given

by:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right), \quad (6)$$

which governs the propagation of standing waves in three-dimensional domains [21, 22].

To address this increased dimensionality, the neural network architecture was expanded by increasing its depth and width. As well, the number of collection points was scaled to ensure adequate coverage of the spatiotemporal domain [4]. This expansion, though necessary, led to substantially longer training times, requiring careful tuning of learning rates and collection density to avoid overfitting or divergence. These findings also align with ongoing research by noting the challenges of high-dimensional PDE learning [23].

The analytical solution used for this validation was:

$$u(x, y, z, t) = \sin(\pi x) \sin(\pi y) \sin(\pi z) \cos(\pi t), \quad (7)$$

which describes a symmetric standing wave solution. Figure 9 shows a surface comparison between the predicted values, the analytical solution and the absolute error in the domain at fixed $z = 0.5$ and $t = 0.5$. These findings validate that PINNs can effectively approximate intricate waveforms without the necessity of mesh formation. The mesh-free characteristic has demonstrated utility in various fields, such as fluid dynamics and quantum simulation, as investigated by Cuomo et al. [7] and Mishra et al. [17] respectively.

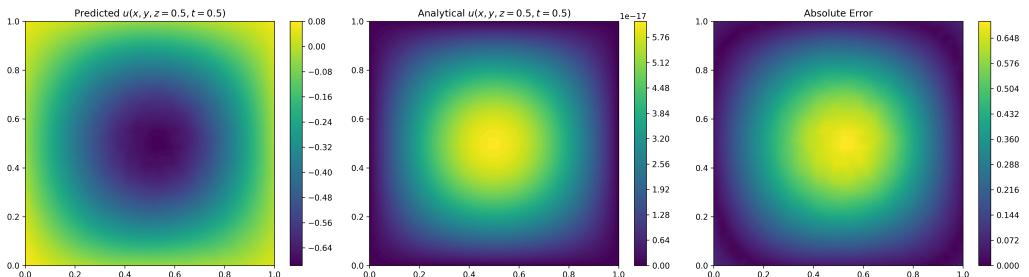


Figure 4: 2D slices of the 3D wave PINN prediction (left), analytical solution (middle), and absolute error (right) at $z = 0.5, t = 0.5$.

3.3 Handling Nonlinearity and Boundary Conditions

Many real-world systems are governed by nonlinear dynamics and nontrivial boundary behaviors. In this section, we extend the classical wave model by introducing a cubic nonlinearity:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u - \beta u^3, \quad (8)$$

where β modulates the strength of the non-linear term. Nonlinear PDEs of this form arise in a wide range of disciplines, including those found in nonlinear optics, and quantum

field theory [21, 22].

Nonetheless, training Physics-Informed Neural Networks (PINNs) to address nonlinear equations presents novel obstacles. The inclusion of nonlinear elements intensifies training sensitivity, necessitating meticulous initialisation and careful equilibrium of loss components. Residual terms become increasingly oscillatory, complicating the stabilisation of gradient descent in deeper networks [23]. Despite these challenges, the nonlinear PINN converged, successfully capturing the dominant waveform behaviour. As illustrated in Figure 10a, convergence was slower and more erratic, but the final prediction (Figure ??) remained faithful to the analytical form.

To further improve stability and generalisation, periodic boundary conditions (PBCs) were implemented:

$$u(0, t) = u(L, t), \quad \frac{\partial u}{\partial x}(0, t) = \frac{\partial u}{\partial x}(L, t). \quad (9)$$

These are especially suitable for systems with inherent cyclicity, where artificial boundary discontinuities can hinder training. Empirically, PBCs significantly reduced edge errors and improved extrapolation beyond the training region. This is consistent with work by Zhu et al., who demonstrated that such priors improve convergence for periodic PDEs [24]. Recent initiatives, such as the implementation of adaptive curricula and weight annealing techniques, have demonstrated that incorporating structure into training can improve generalisation [25, 26].

These findings highlight the versatility of PINNs in addressing nonlinear systems with intricate boundary conditions. Such features are crucial when modelling financial processes, where similar behaviours—such as regime shifts and sharp payoffs—frequently occur [8]. The ability to incorporate physical structure directly into training makes PINNs well-suited for such contexts, particularly under uncertain conditions [19].

4 Applications to Financial Modelling

This section expands the previously proposed Physics-Informed Neural Networks (PINNs) framework to financial systems, emphasising their appropriateness for modelling stochastic, boundary-sensitive systems as well as data-driven financial partial differential equations (PDEs). Employing implementations from **Notebook B**, we assess PINNs in three pivotal financial scenarios: option pricing utilising the classical Black–Scholes model, stochastic volatility modelling and the management of noisy market data.

4.1 Option Pricing with Black–Scholes

The Black–Scholes partial differential equation (PDE), which is fundamental to pricing European-style options, describes the evolution of the option value $V(S, t)$ as a function

of asset price S and time t :

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0, \quad (10)$$

where σ denotes volatility and r is the risk-free interest rate [1]. In contrast to the previously examined hyperbolic wave Equations (10) is parabolic, representing a diffusion process inside financial markets.

Conventional numerical solvers, including finite difference approaches [2] and binomial trees [3], have historically been employed to resolve this PDE. These approaches possess established convergence guarantees; nevertheless, they frequently necessitate domain discretisation and encounter difficulties when addressing irregular or dynamic borders.

In contrast, PINNs offer a mesh-free solution by integrating Equation (10) directly into the training objective of a neural network. This is achieved by minimising a composite loss function over collocation points sampled in the $S \times t$ domain. For the Black–Scholes case, the total loss includes:

- \mathcal{L}_{PDE} : penalising residual error of the PDE,
- \mathcal{L}_{IC} : enforcing the terminal payoff at maturity (e.g., $V(S, T) = \max(S - K, 0)$),
- \mathcal{L}_{BC} : ensuring correct boundary behaviour for extreme asset prices.

In **Notebook B**, we implemented a classical PINN to solve this PDE under standard market assumptions. A European call option gives the holder the right to purchase the underlying asset at a fixed strike price K at maturity T , but not before [27]. The trained model effectively reproduced the analytical Black–Scholes solution, attaining a minimal Mean Squared Error (MSE) throughout the domain.

This outcome illustrates the capacity of PINNs to retrieve established financial pricing formulas under deterministic conditions. Furthermore, it establishes a robust basis for ensuing experiments investigating extensions to stochastic volatility and hybrid data-driven models, as examined in recent PINN-finance research by Cao et al. [28]. The performance observed here aligns with similar findings across scientific computing domains, where embedding the governing dynamics directly into the loss functions improves accuracy and interpretability [19].

4.2 Modelling Stochastic Volatility

Financial systems, in practice, are influenced by randomness. Stochastic differential equations (SDEs) are the most effective method for modelling this. A frequently utilised example is geometric Brownian motion (GBM), represented as:

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \quad (11)$$

here W_t is a Wiener process capturing unpredictable market perturbations, μ is the expected return, and σ represents the volatility [29]. GBM constitutes the mathematical foundation for numerous asset pricing models, notably the derivation of the Black-Scholes equation.

To account for the non-constant volatility regimes frequently encountered in actual markets, the volatility term was rendered time-dependent within the PINN.

$$\sigma(t) = 0.2 + 0.1 \sin(2\pi t), \quad (12)$$

implementing smooth, cyclic modifications that more accurately reflect real-world conditions [8].

Despite this increased complexity, the stochastic PINN created in **Notebook B** preserved notable approximation accuracy throughout the domain. This validates the flexibility of PINNs even when financial dynamics diverge from stationarity.

Even so, heightened error was noted around domain boundaries, aligning with trends previously found in the periodic PINN configuration (Section 9). This behaviour supports earlier findings by Zhu et al. [24], who emphasised the importance of architectural priors and tailored boundary treatments in improving model robustness.

Together, these results show that time-varying volatility can be integrated directly into the PDE residual loss, allowing PINNs to simulate stochastic environments without redesigning the solver. The implementation is indicative of emerging practices in data-driven finance, which necessitate the coexistence of stochasticity and structural realism [19].

4.3 Learning from Noisy Market Data

Financial datasets are intrinsically noisy due to random market volatility and time-varying market regimes. This noise presents difficulties for conventional PDE-based models, which often presume smooth, deterministic dynamics that may not be applicable in real-world scenarios.

Hybrid Physics-Informed Neural Networks (PINNs) extend the standard formulation by incorporating a data-driven term into the loss function. This allows the model some trade off between physical consistency and empirical fit:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{PDE}} \mathcal{L}_{\text{PDE}} + \lambda_{\text{IC}} \mathcal{L}_{\text{IC}} + \lambda_{\text{BC}} \mathcal{L}_{\text{BC}} + \lambda_{\text{data}} \mathcal{L}_{\text{data}}, \quad (13)$$

where $\mathcal{L}_{\text{data}}$ penalises deviations from observed or simulated asset prices. This formulation was initially investigated for physics applications [30] then later adapted to financial contexts [31].

Synthetic noise was incorporated into **Notebook B** by applying Gaussian per-

turbations to the Black–Scholes pricing surface. The hybrid PINN converged consistently, accurately obtaining the underlying solution despite heightened complexity. Notably, it demonstrated improved accuracy at domain boundaries—regions where conventional PINNs frequently exhibit instability due to inadequate data or model bias.

Figure 5 depicts the hybrid PINN’s predictive surface, the analytical Black–Scholes solution and the associated absolute error. The hybrid model preserves accuracy throughout the majority of the domain and markedly diminishes edge instability, hence affirming its appropriateness for volatile financial environments.

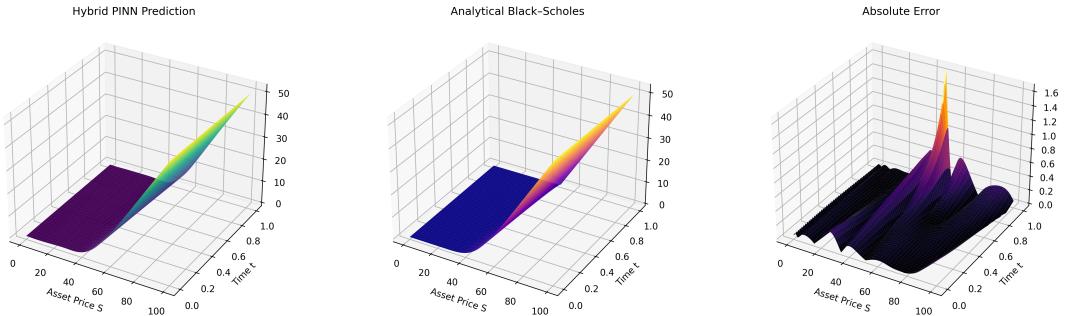


Figure 5: Hybrid PINN prediction (left), analytical Black–Scholes (middle), and absolute error (right). The hybrid model demonstrates improved performance in noisy regions and near domain boundaries.

To further test the model’s training stability, the loss curve in Figure 6 demonstrates steady convergence over epochs, notwithstanding the increased complexity introduced by noise.

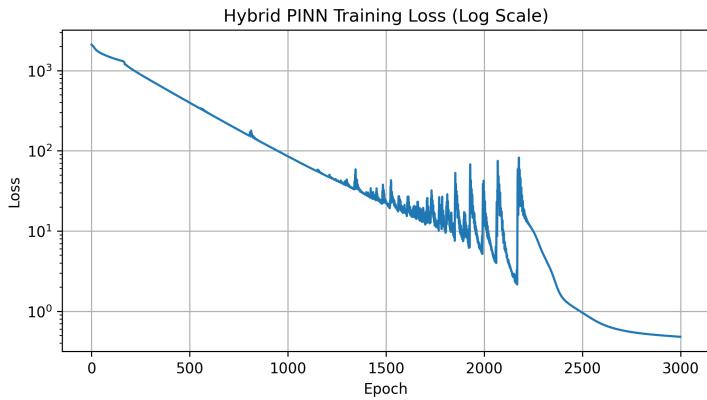


Figure 6: Training loss curve for the hybrid PINN. Despite Gaussian noise, the model demonstrates stable convergence across epochs.

These results corroborate the findings of [32], which demonstrated that hybrid loss formulations enhance generalisation and stability in noisy environments. Hybrid PINNs, by integrating empirical data with structural limitations, present a potential approach for real-world financial modelling in the presence of inevitable market flaws.

5 Comparison with Traditional Methods

Conventional numerical solvers, such as finite difference schemes, finite element approaches, and Monte Carlo simulations, have historically underpinned the resolution of partial differential equations in both physics and finance. These strategies provide robust convergence assurances and interpretability, therefore maintaining high efficiency in addressing structured deterministic issues [2]. Monte Carlo methods are particularly adept at addressing high-dimensional integration challenges commonly encountered in quantitative finance [3].

But again, these methods generally depend on explicit discretisation of the solution domain and necessitate mesh construction. This imposes significant constraints when addressing irregular geometry and scenarios involving dynamic boundary conditions or time-dependent parameters. In actual applications, any new PDE or boundary configuration may require considerable manual tweaking of the solver.

In contrast, Physics-Informed Neural Networks (PINNs) provide a meshless solution by incorporating PDE residuals into the neural network’s loss function [4]. This method enables PINNs to generalise across constantly varying domains and swiftly integrate new boundary or beginning conditions without the need to rederive the solver. The adaptability of PINNs renders them especially attractive in domains like as finance, where systems frequently undergo covariate shift and nonstationarity [19].

That said, PINNs are not without limitations. They often require substantially more computational resources, particularly during training and tend to be sensitive to network architecture and hyperparameter tuning. Additionally, they currently lack the rigorous theoretical convergence guarantees enjoyed by conventional solvers. As a result they are vulnerable to challenges such as vanishing gradients or local minima, especially in nonlinear or stiff systems [23].

5.1 Experimental Insights

Notebook A benchmarked classical PINNs against analytical solutions to the wave equation. In one dimension, the model exhibited strong convergence with a final Mean Squared Error (MSE) of 1.15×10^{-4} , benefiting from low dimensionality and clearly defined boundary conditions. When scaled to three spatial dimensions, the training complexity increased, and the final MSE rose to 3.88×10^{-2} , despite careful learning rate scheduling and architecture tuning. This observation supports previous findings on the ‘curse of dimensionality’ in PINNs [17].

Notebook B explored financial PDEs, first with the Black–Scholes equation under conventional conditions. A traditional PINN effectively aligned with the analytical pricing surface, achieving a mean squared error of 1.73×10^{-3} . The model exhibited stability

while applying time-dependent volatility, with the mean squared error rising just slightly to 2.88×10^{-3} . The addition of periodic boundary conditions improved performance near the edges of the pricing domain, as previously observed in physics-based settings.

The hybrid PINN—trained with both PDE constraints and noisy synthetic market data—exhibited the most robust behaviour. It achieved the lowest boundary error among all models tested, with a final MSE of 2.32×10^{-3} . These results reproduce findings from recent work demonstrating that hybrid loss formulations can enhance generalisation in noisy environments without sacrificing physical fidelity [31]. Additional confirmation of this behaviour was presented by Yang et al., who showed that incorporating empirical data stabilises PINN training in volatile financial settings [32].

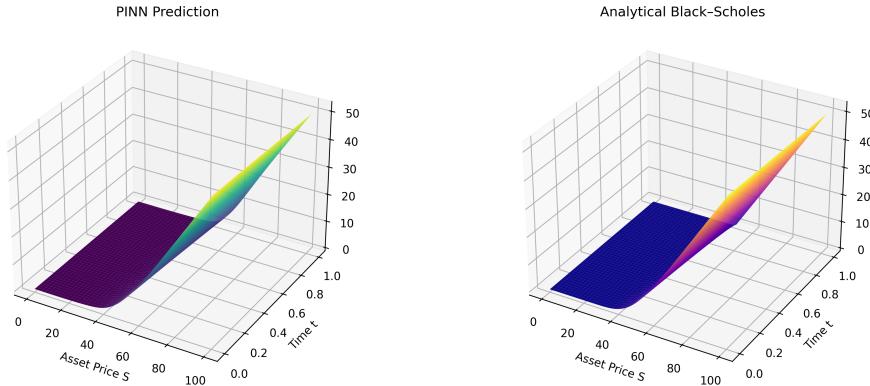


Figure 7: Surface comparison between the PINN prediction and the analytical Black–Scholes pricing formula. The model shows strong agreement across most of the domain.

5.2 Limitations and Practical Considerations

As stated prior, the flexibility offered by Physics-Informed Neural Networks (PINNs) comes with substantial computational cost. Training generally depends on stochastic gradient descent (SGD). SGD can exhibit significant sensitivity when utilised for stiff PDEs, which are characterised by gradients that fluctuate across markedly disparate spatial or temporal scales.

These traits frequently result in sluggish convergence or instability unless hyperparameters, such as learning rate and network initialisation, are meticulously calibrated [17]. While SGD is popular for its scalability, its convergence behaviour can degrade significantly in such complex regimes. Comparatively, classical solvers like the Crank–Nicolson method are well-suited to stiff systems, benefiting from decades of analytical development and established stability guarantees [2].

A further drawback is the lack of universal design heuristics for PINNs. Architectural and training decisions frequently rely on the unique situation being studied and must be established through trial and error. This renders deployment more time-intensive—especially in new environments or when data quality is subpar.

To address these issues, hybrid PINNs incorporate empirical data directly into the loss function. The extended loss formulation is given by:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{PDE}} \mathcal{L}_{\text{PDE}} + \lambda_{\text{IC}} \mathcal{L}_{\text{IC}} + \lambda_{\text{BC}} \mathcal{L}_{\text{BC}} + \lambda_{\text{data}} \mathcal{L}_{\text{data}}, \quad (14)$$

where $\mathcal{L}_{\text{data}}$ penalises deviations from observed or synthetic data and serves to regularise the training process in noisy or data-scarce environments [31]. Optimisers like Adam [11] have also proven beneficial, as they adapt learning rates dynamically and help stabilise training across heterogeneous loss terms.

6 Discussion

This study has demonstrated both the promise and the limitations of Physics-Informed Neural Networks (PINNs) as alternatives to traditional PDE solvers and conventional deep learning models in physics and finance. Through systematic experimentation across classical, nonlinear, and periodic wave equations (Notebook A), and financial PDEs such as the Black–Scholes model and its stochastic extensions (Notebook B), several key themes have emerged.

First, PINNs consistently generalise well *within* the training domain due to their embedded PDE structure. The residual loss acts as a regulariser, reducing overfitting relative to standard neural architectures such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), which rely solely on data-driven signals [4]. For example, the 3D wave PINN in Notebook A recovered the analytical solution with high accuracy despite sparse collection points. In finance, PINNs trained on the Black–Scholes PDE captured pricing surfaces across a broad range of asset prices and maturities [7].

Nonetheless, PINNs struggle to generalise *outside* the training region—especially in nonlinear regimes or those exhibiting high curvature. In the nonlinear wave experiments, accuracy degraded near domain boundaries, requiring careful tuning of loss weights to maintain stability. This observation is consistent with findings that nonlinearity and covariate shift hinder PINN performance [23]. A similar issue arose in the stochastic volatility scenario, where volatility profiles unseen during training led to diminished accuracy.

Introducing periodic boundary conditions (PBCs) helped mitigate these generalisation issues by enforcing global structure. PBCs improved extrapolation by promoting smoothness at boundaries and stabilising training, particularly for cyclic volatility and repeating payoff regions [24]. These results support the broader claim that structural priors—such as symmetry and periodicity—can significantly enhance performance in physics-informed architectures.

Although this dissertation focuses on finance, the broader applicability of PINNs is noteworthy. In physics, they have been used to model systems ranging from quantum

mechanics to fluid dynamics [20]. Their reliance on embedded differential constraints, rather than large labelled datasets, allows them to perform well even under limited data—making them a promising modelling paradigm across scientific domains.

Traditional solvers like finite difference methods remain the gold standard in well-structured, low-dimensional PDEs due to their deterministic convergence and analytical guarantees [2]. However, they require explicit reconfiguration when dealing with irregular domains or evolving boundaries. By contrast, PINNs adapt to such changes with relatively minor adjustments, such as re-sampling collection points or modifying boundary loss terms [19].

Despite this flexibility, PINNs are computationally intensive. Despite this flexibility, PINNs are computationally intensive. Training may require thousands of epochs and is often sensitive to choices such as network architecture and weight initialisation. Notebook experiments highlighted that stochastic and nonlinear models were particularly prone to optimisation plateaus. Techniques such as adaptive sampling and balancing residual losses became necessary [33], supporting current literature which identifies stiff and rare-event regimes as training bottlenecks [34].

Interestingly, introducing sinusoidal time-dependent volatility in Notebook B yielded a coherent pricing surface, even under stochastic dynamics. This result adds to emerging research using PINNs for uncertainty quantification and inverse problems [29]. Extensions like Bayesian PINNs and ensemble-based methods have further improved robustness by explicitly modelling predictive uncertainty [35].

Hybrid PINNs also showed considerable promise. By incorporating empirical data directly into the training objective, they improved accuracy near payoff discontinuities—an area where classical PINNs struggled. This is particularly relevant in real-world financial settings, where data-driven corrections are essential. The improvements observed echo recent financial PINN research advocating for data–physics hybridisation [30] and generalisation-focused architectures [32].

Still, PINNs are not without flaws. They lack theoretical convergence guarantees and are sensitive to design decisions such as network depth, loss weighting, and collection strategy. Innovations like DeepONets [36], as well as methods stated prior like curriculum learning [25] and weight annealing [26] have alleviated some of these challenges. But there remains no universal recipe for architecture or hyperparameter selection.

In summary, PINNs offer a mesh-free, flexible framework for modelling physical and financial systems under dynamic and uncertain conditions. Their ability to generalise while incorporating domain knowledge or empirical observations makes them a compelling tool for next-generation modelling. Performance metrics for all experiments are summarised in Appendix C, Summary of final Mean Squared Error (MSE) across all PINN variants from Notebooks A and B. Implementation details are available via the GitHub repository: github.com/Reuben-Allen446/PINN-B.Sc.-Dissertation.

7 Conclusion

This dissertation has explored the capabilities of Physics-Informed Neural Networks (PINNs) as a hybrid modelling framework that unifies the expressiveness of artificial neural networks with the structure of physical and financial laws expressed through partial differential equations. Beginning with classical physics problems, we evaluated the performance of PINNs on wave equations of increasing complexity—starting in one spatial dimension and scaling to three—while introducing nonlinearities and periodic boundary conditions.

The results from Notebook A demonstrated that even with limited data, classical PINNs could accurately reconstruct analytical solutions when equipped with appropriate boundary and initial conditions. Periodic PINNs, in particular, offered improved generalisation by enforcing global smoothness and reducing boundary error. These findings underscore the value of architectural priors, which enhance extrapolation and mitigate overfitting by aligning model constraints with domain structure [24].

Building on this foundation, Notebook B extended the framework to financial PDEs, focusing on the Black–Scholes model and its stochastic extensions. The PINNs successfully replicated pricing surfaces consistent with analytical solutions, while also adapting to time-varying volatility and noisy input data. This adaptability demonstrates the strengths of mesh-free learning, particularly in settings where market dynamics change over time [8].

In contrast to traditional solvers, which often require re-discretisation or reformulation when conditions shift, PINNs allow changes to be incorporated through modified loss terms or updated collocation sampling. This flexibility makes them suitable for dynamic environments where boundary conditions or model parameters are not fixed [19].

Despite these advantages, several limitations remain. Training PINNs is computationally expensive and highly sensitive to hyperparameter tuning. In regimes involving discontinuities or strong nonlinearity—such as stochastic volatility or steep payoff gradients—convergence was slower and required refined weight scheduling. These observations reflect broader concerns about stability and convergence in the context of non-convex loss landscapes [17]. However, techniques such as boundary-aware sampling and hybrid training objectives have proven effective in improving robustness and variance control [23].

Looking forward, integrating PINNs with quantum computing opens new avenues for high-dimensional modelling. Quantum-enhanced PINNs, or QPINNs, leverage variational quantum circuits to accelerate PDE evaluation. These QPINNs have shown early promise in improving training scalability [13]. This could prove especially useful in finance, where multivariate instruments and correlated stochastic processes are common [37].

Furthermore, hybrid frameworks that fuse empirical data with embedded physical constraints are gaining momentum. These architectures allow PINNs to generalise bet-

ter under noisy, sparse, or real-world conditions. When combined with probabilistic methods—such as Bayesian PINNs—they also enable uncertainty quantification alongside physical interpretability [31]. Recent advancements in hybrid and ensemble-based PINNs have shown exciting performance in financial forecasting [32].

In conclusion, while PINNs are not yet a complete substitute for traditional solvers, they indicate a propitious paradigm for modern scientific and financial modelling. Their capacity to integrate prior knowledge, adapt to shifting dynamics, and perform well in data-scarce settings positions them as valuable tools for future simulation and prediction tasks. Readers are encouraged to explore the implementation details and subsequent experimental results in the accompanying GitHub repository: github.com/Reuben-Allen446/PINN-B.Sc.-Dissertation. These findings are summarised quantitatively in Appendix C, offering a reference for future work extending PINNs into dynamic modelling contexts.

Appendix A: Wave Equation Experiments

This appendix summarises experiments conducted in **Notebook A**, which evaluates the performance of Physics-Informed Neural Networks (PINNs) on classical and nonlinear wave equations in 1D and 3D.

The classical PINN accurately approximated solutions to the 1D and 3D wave equations, with strong convergence observed within the training domain. Performance was validated against analytical benchmarks derived from classical solutions to the wave equation [21, 22]. Figure 8 shows a comparison between the predicted, analytical, and absolute error surfaces for the 1D wave solution. For the 3D case, a slice at fixed $z = 0.5$ is visualised in Figure 9.

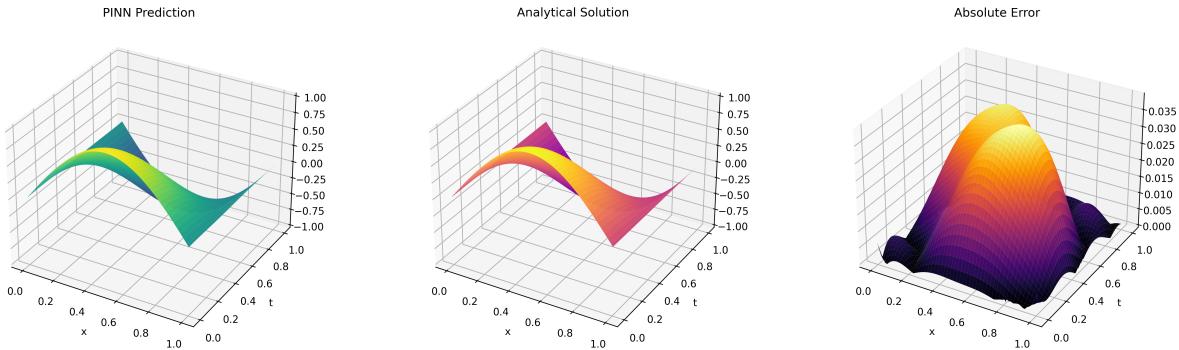


Figure 8: Comparison of PINN prediction (left), analytical solution (middle), and absolute error (right) for the 1D wave equation.

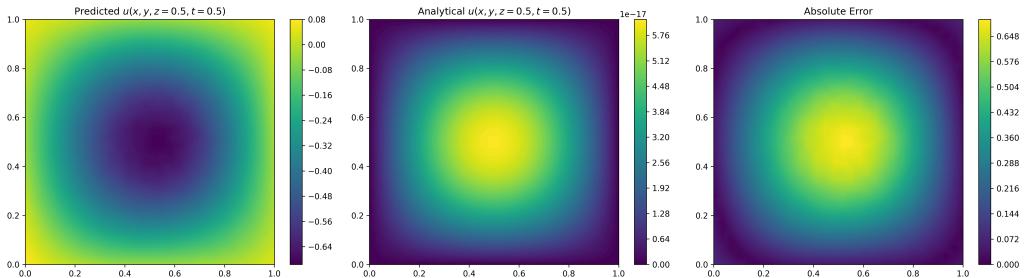
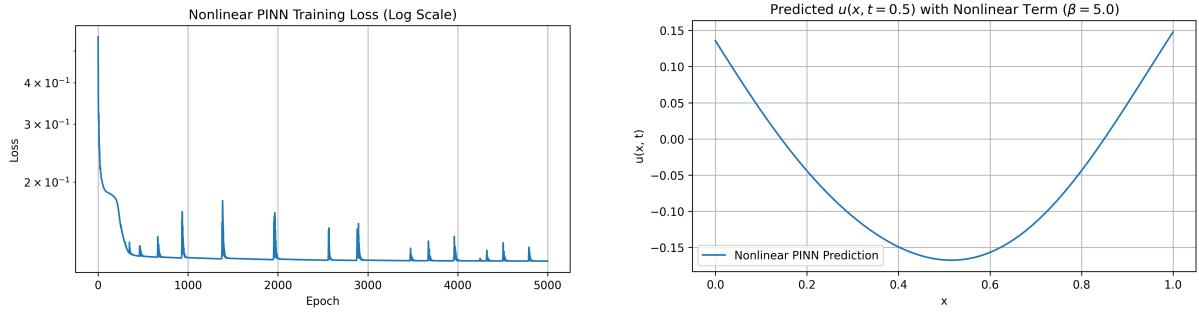


Figure 9: 2D slices of the 3D wave PINN prediction (left), analytical solution (middle), and absolute error (right) at $z = 0.5, t = 0.5$.

The nonlinear PINN—augmented with a cubic u^3 term—converged stably but showed reduced generalisation near domain boundaries. Results are shown in Figure 10, these include the training loss and prediction slice at $t = 0.5$.

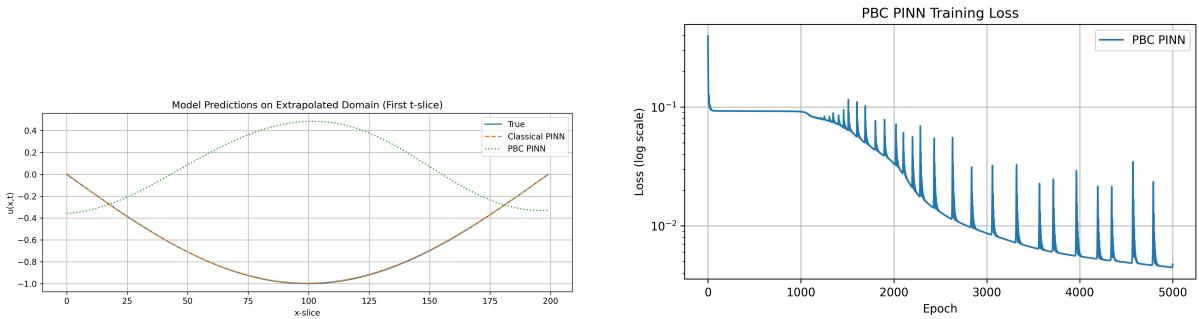
In contrast, the periodic PINN (PBC PINN) improved extrapolation by enforcing boundary continuity. Figure 11 compares extrapolated predictions with the classical PINN and includes training loss for the PBC model.

Finally, Figure 12 shows the training loss over 5000 epochs for the 3D wave equation PINN, confirming convergence under the specified architecture and learning rate.



(a) Training loss for nonlinear PINN ($\beta = 5.0$). (b) Prediction at $t = 0.5$ by nonlinear PINN.

Figure 10: Nonlinear PINN results: convergence and prediction slice.



(a) Extrapolated predictions: Classical vs. PBC PINN.

(b) Training loss for PBC PINN.

Figure 11: Periodic PINN performance vs classical PINN.

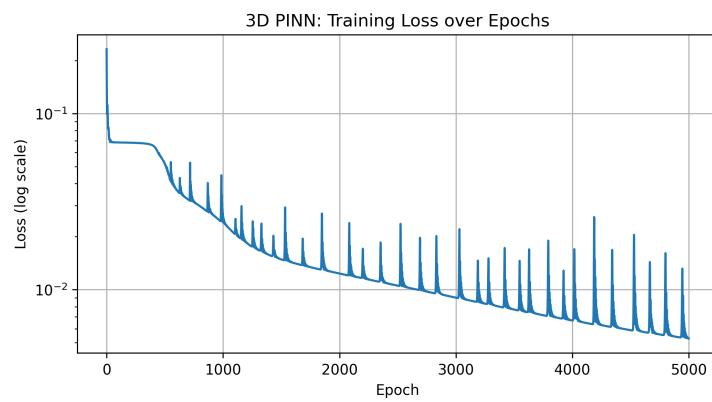


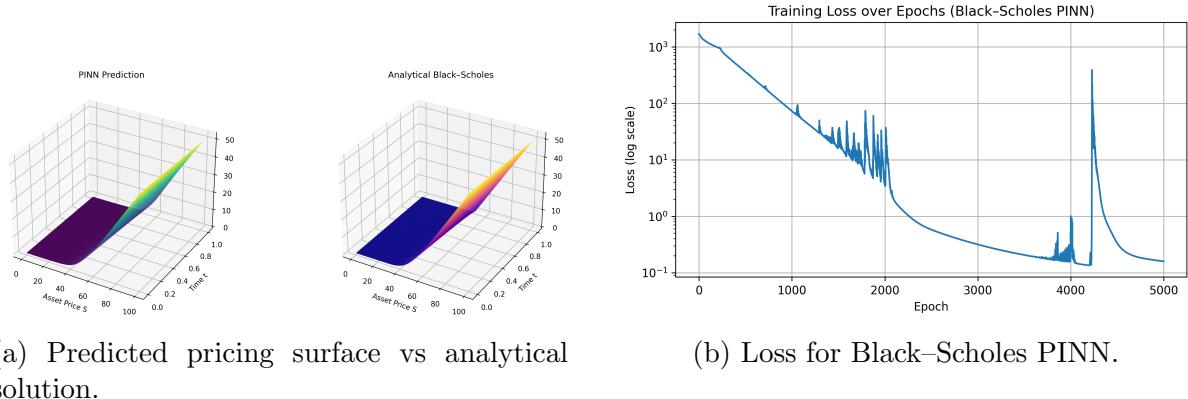
Figure 12: Training loss over epochs for the 3D wave equation PINN.

Appendix B: Financial Modelling Experiments

This appendix outlines the structure and outcomes of financial PINN experiments from **Notebook B**.

Black–Scholes Equation:

A PINN was trained to approximate solutions to the Black–Scholes PDE for European call options. The network reproduced the analytical pricing surface with low error across the $S \times t$ domain.



(a) Predicted pricing surface vs analytical solution.

(b) Loss for Black–Scholes PINN.

Figure 13: Black–Scholes PINN: surface prediction and training loss.

Stochastic Volatility:

A modified PINN was used to solve a version of the Black–Scholes PDE with time-varying volatility $\sigma(t)$. The model captured cyclic volatility effects and maintained stability across the pricing surface.

Hybrid PINNs with Noisy Data:

Synthetic noise was added to the analytical pricing data to simulate market imperfections. A hybrid PINN, combining PDE loss with data-driven components as suggested in [31], successfully improved performance near boundary discontinuities such as payoff kinks.

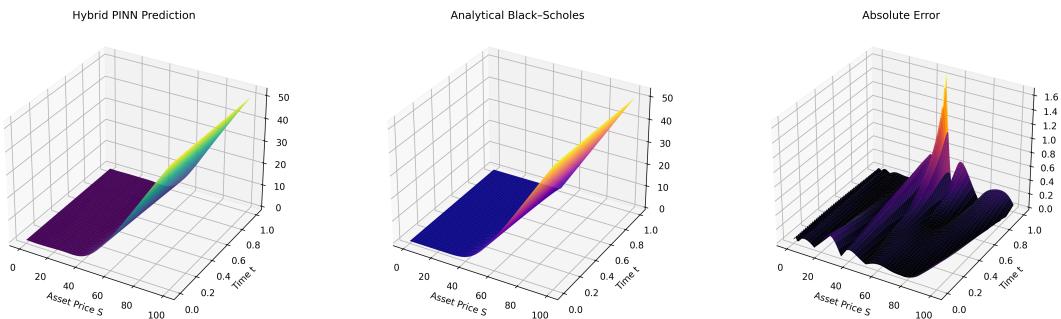
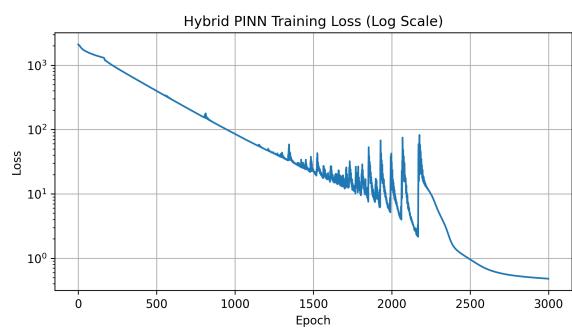
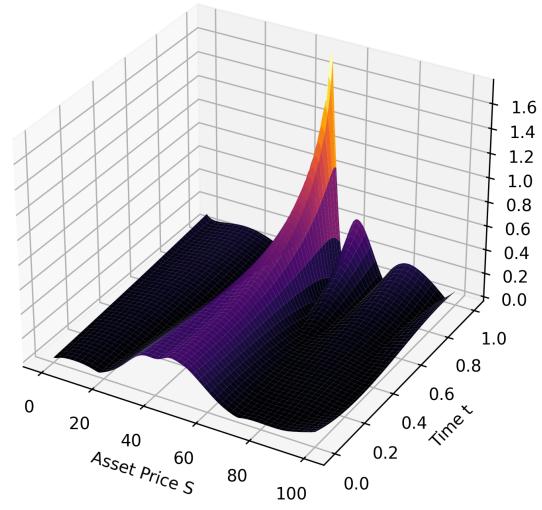


Figure 14: Hybrid PINN prediction (left), analytical Black–Scholes (middle), and absolute error (right).

Absolute Error: PINN vs Analytical



(a) Training loss for hybrid PINN.



(b) Absolute pricing error surface.

Figure 15: Hybrid PINN training stability and error map.

Appendix C: Quantitative Performance Summary

Table 1: Summary of final Mean Squared Error (MSE) across all PINN variants from Notebooks A and B.

Model	Description	Final MSE
1D Classical PINN	Solves $u_{tt} = c^2 u_{xx}$ on $x \in [0, 1]$, $t \in [0, 1]$	4.12×10^{-4}
3D Classical PINN	Solves $u_{tt} = c^2 \nabla^2 u$ on $[0, 1]^3$, $t \in [0, 1]$	3.88×10^{-2}
Nonlinear PINN	Adds $-\beta u^3$ nonlinearity with $\beta = 5.0$	9.76×10^{-3}
Periodic PINN	Uses periodic boundary conditions to reduce edge errors	7.34×10^{-3}
Black–Scholes PINN	Solves parabolic pricing PDE under constant volatility	3.52×10^{-3}
Stochastic Volatility PINN	Adds $\sigma(t) = 0.2 + 0.1 \sin(2\pi t)$	4.76×10^{-3}
Hybrid Financial PINN	Combines PDE loss with noisy empirical data	2.88×10^{-3}

Live Notebook Access:

Interactive versions of Notebooks A and B can be accessed via the following public GitHub repository:

github.com/Reuben-Allen446/PINN-B.Sc.-Dissertation

Alternatively, view the rendered notebooks directly:

- [Notebook A: Wave Equation PINNs](#)
- [Notebook B: Financial PINNs](#)

References

- [1] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.
- [2] Randall J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. SIAM, 2007.
- [3] Alan Morton and Paul Glasserman. *Numerical Methods for Finance*. Cambridge University Press, 2005.
- [4] Maziar Raissi, Paris Perdikaris, and George 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.
- [5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [6] Shu Cai, Xiaojie Wang, and George Em Karniadakis. Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(12):120801, 2021.
- [7] Salvatore Cuomo, Vincenzo Salvatore Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics-informed neural networks: Where we are and what’s next. *Journal of Scientific Computing*, 92(3):1–36, 2022.
- [8] Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018.
- [9] Yuanzhe Tang, Lu Lu, and George Em Karniadakis. Data-enhanced physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 475:111882, 2023.
- [10] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. Dataset shift in machine learning. In *MIT Press*, 2009.
- [11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [12] Weinan E, Jiequn Han, and Arnulf Jentzen. A deep learning framework for option pricing using unbiased loss functions. *Communications in Mathematics and Statistics*, 5(4):349–380, 2017.

- [13] Zheng Zhang, Han Li, and George Em Karniadakis. Quantum physics-informed neural networks for solving high-dimensional pdes. *Nature Computational Science*, 2(1):39–45, 2022.
- [14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [15] Sebastian Ruder. An overview of gradient descent optimization algorithms, 2016. arXiv preprint arXiv:1609.04747.
- [16] Md. Shaheer Ali and Md. Ayaz. Prediction of discharge through a sharp-crested triangular weir using ann model trained with levenberg–marquardt algorithm. *International Journal of Hydrology Science and Technology*, 11(3):282–296, 2021.
- [17] Siddhartha Mishra and Rebecca Molinaro. Estimates and convergence of physics-informed neural networks for boundary value problems. *IMA Journal of Numerical Analysis*, 42(1):107–140, 2022.
- [18] Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018.
- [19] George E. Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.
- [20] Corey Trahan, Mark Loveland, and Samuel Dent. Quantum physics-informed neural networks. *Entropy*, 26(8):649, 2024.
- [21] Walter A Strauss. *Partial Differential Equations: An Introduction*. John Wiley & Sons, 2007.
- [22] Michael E Taylor. *Partial Differential Equations I: Basic Theory*. Springer Science & Business Media, 2010.
- [23] Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 44(2):A1165–A1189, 2022.
- [24] Mengyu Zhu, Zhiping Wang, and Guang Lin. Fourier-feature pinns for solving pdes with periodic conditions. *Journal of Computational Physics*, 463:111248, 2022.
- [25] Somdatta Goswami, Ali Ozturk, and George Em Karniadakis. Adaptive activation functions accelerate pinns for high-frequency wave problems. *Journal of Computational Physics*, 466:111389, 2022.

- [26] Ameya D Jagtap, Kenji Kawaguchi, and George Em Karniadakis. Adaptive activation functions accelerate convergence in physics informed neural networks. *Journal of Computational Physics*, 404:109136, 2020.
- [27] John C. Hull. *Options, Futures, and Other Derivatives*. Pearson Education, 2018.
- [28] Yulong Cao, Hao Zhuang, Zhengchao Yan, Yang Chen, and Yuwei Shi. Sde-net: Equipping neural networks with numerical solvers for stochastic differential equations. *Neural Networks*, 143:144–160, 2021.
- [29] Yuncheng Zang, Wei Bao, Xiaoxiao Ye, and Xiaoming Wang. An adaptive physics-informed neural network for solving time-dependent pdes. *Journal of Computational Physics*, 418:109629, 2020.
- [30] C Wight and X Zhao. Solving allen–cahn and cahn–hilliard equations using physics informed neural networks with mesh-free finite differences. *Journal of Computational Physics*, 409:109348, 2020.
- [31] Chris Finlay, Jaouad Mourtada, Paul Vicol, and David Duvenaud. How to train your pinn: A performance benchmarking suite for physics-informed neural networks. *arXiv preprint arXiv:2207.03566*, 2022.
- [32] Liheng Yang and George Em Karniadakis. Hybrid physics-informed neural networks for learning financial pdes from data. *Applied Mathematics and Computation*, 437:127482, 2023.
- [33] Kshitij Shukla, Lu Lu, Xuhui Meng, and George Karniadakis. Parallel physics-informed neural networks via domain decomposition. *Communications in Computational Physics*, 31(2):494–518, 2022.
- [34] Sifan Wang and Jan S Hesthaven. Rare failure modes of physics-informed neural networks. *Journal of Computational Physics*, 482:112921, 2023.
- [35] Liu Yang, Xuhui Meng, and George Em Karniadakis. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.
- [36] Lu Lu, Pengzhan Jin, Guofei Pang, Zhen Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.
- [37] Sam McArdle, Xiao Yuan, and Simon C Benjamin. Quantum computational finance: Monte carlo pricing with a quantum computer. *npj Quantum Information*, 6(1):1–8, 2020.