Al-Qadisiyah Journal of Pure Science

ISSN(Printed): 1997-2490 ISSN(Online): 2411-3514

DOI: 10.29350/jops

Al-Qadisiyah Journal of Pure Science

Efficiently Solving Complex PDEs with DRM-QMC Algorithm

Suhad Talib Yousif

Faculty of Applied Mathematics, department of numerical analysis, Islamic Azad University, Iran.

*Corresponding author email: suhads862@gmail.com

Abstract

The Deep Ritz Method - Quasi-Monte Carlo (DRM-QMC) algorithm represents a novel approach that integrates deep learning and Quasi-Monte Carlo sampling for efficiently solving complex partial differential equations (PDEs). By leveraging neural networks for solution approximation and QMC sampling for point generation, the algorithm aims to provide accurate solutions with improved convergence rates for high-dimensional PDEs. The DRM-QMC algorithm demonstrates promising potential in accurately solving complex PDEs with enhanced convergence properties. Through numerical experiments and convergence analysis, the algorithm showcases improved accuracy and convergence rates, highlighting the effectiveness of deep learning methods in tackling high-dimensional PDEs. The comparison of total errors provides valuable insights into the algorithm's performance and reliability in solving intricate PDEs. Regularization techniques and QMC sampling are identified as key factors influencing the convergence properties of the DRM-QMC algorithm. By optimizing these components, researchers can enhance the algorithm's efficiency and reliability in approximating solutions to high-dimensional PDEs. The equation $E(\theta) \leq C(\log f(N))\hat{}\alpha N\hat{}-\beta$ serves as a quantitative measure for assessing the algorithm's convergence properties and guiding future advancements in deep learning methods for PDEs. Overall, the DRM-QMC algorithm represents a significant advancement in computational mathematics, offering a promising approach for efficiently solving complex PDEs with improved accuracy and convergence rates.

Keywords: DRM, QMC, PDEs, Convergence Analysis, and Deep Learning.

1. Introduction

Partial differential equations (PDEs) are fundamental mathematical tools for modeling various physical phenomena, such as fluid dynamics, heat transfer, and electromagnetism [1]. However, solving high-dimensional PDEs remains a challenging task due to the curse of dimensionality. In recent years, deep learning has emerged as a promising approach for solving PDEs, thanks to its ability to approximate complex functions and capture intricate patterns in high-dimensional spaces. Let us consider a general PDE of the form:

$$Lu(x)=f(x),x\in\Omega Lu(x)=f(x),x\in\Omega$$

where LL is a differential operator, $u(x)$u(x) is the unknown solution, $f(x)$f(x) is a given function, and $\Omega\Omega$ is the domain of interest.

Deep learning algorithms aim to approximate the solution $u(x)$u(x) by training a neural network $u\theta(x)$uθ(x), where $\theta\theta$ represents the network parameters. The loss function for these PDE problems is often defined as an integral over the domain $\Omega\Omega$:

$$J(\theta)=\int\Omega|Lu\theta(x)-f(x)|2dxJ(\theta)=\int\Omega|Lu\theta(x)-f(x)|2dx$$

The goal is to find the optimal parameters $\theta*\theta*$ that minimize the loss function:

$$\theta*=\arg f()\min f()\theta J(\theta)\theta*=\arg\theta\min J(\theta)$$

Convergence analysis plays a crucial role in understanding the theoretical properties and practical performance of deep learning algorithms for solving PDEs. It provides insights into the rate at which the approximation error decreases as the number of training samples or network parameters increases. Convergence analysis also helps establish the stability and robustness of the algorithms under various assumptions and conditions [2]. Let $u(x)$u(x) be the true solution and $u\theta(x)$uθ(x) be the neural network approximation. The approximation error can be defined as:

$$E(\theta)=\|u-u\theta\|L2(\Omega)E(\theta)=\|u-u\theta\|L2(\Omega)$$

where $\|\cdot\|L2(\Omega)\|\cdot\|L2(\Omega)$ denotes the $L2$L2-norm over the domain $\Omega\Omega$.

Quasi-Monte Carlo (QMC) sampling has been used to improve the efficiency of deep learning algorithms for PDEs. QMC methods generate deterministic sequences of points that exhibit low discrepancy, leading to faster convergence rates compared to random sampling [3]. The loss function in Equation (2) can be approximated using QMC sampling as:

$$J(\theta)\approx1N\sum i=1N|Lu\theta(xi)-f(xi)|2J(\theta)\approx N1i=1\sum N|Lu\theta(xi)-f(xi)|2$$

where $\{xi\}i=1N$ {xi}i=1N are the QMC sample points.

The objective of this study is to provide a convergence analysis of a quasi-Monte Carlo-based deep learning algorithm for solving PDEs. Under certain assumptions on the regularity of the solution and the properties of the QMC sequence, it can be shown that the approximation error satisfies:

$$E(\theta)\leq C(\log^{f0}N)\alpha N-\beta \text{E}(\theta)\leq \text{C}(\log\text{N})\alpha\text{N}-\beta$$

where CC is a constant, $\alpha\alpha$ and $\beta\beta$ are positive real numbers that depend on the dimension of the problem and the smoothness of the solution, and NN is the number of QMC sample points.

2. QMC methods in DRM

Quasi-Monte Carlo (QMC) methods are a class of numerical integration techniques that aim to improve upon the convergence rate of standard Monte Carlo methods. Unlike traditional Monte Carlo sampling, which uses pseudo-random numbers, QMC methods utilize low-discrepancy sequences to generate deterministic samples that are more evenly distributed over the integration domain [4].

The key advantage of QMC methods is that they can achieve a convergence rate proportional to $N-\beta$N−β, where NN is the number of sample points and $\beta\beta$ is a positive real number that depends on the smoothness of the integrand and the dimension of the problem [5]. This is a significant improvement over the $N-1/2$N−1/2 rate of standard Monte Carlo integration.

Deep learning has been successfully applied to the numerical solution of partial differential equations (PDEs), where the loss function is often defined as an integral over the domain. In this context, QMC methods can be used to approximate the loss function more efficiently than standard Monte Carlo sampling [1].

The main idea is to replace the Monte Carlo integration in the function with a QMC-based approximation. This can lead to faster convergence of the deep learning algorithm and improved accuracy in the PDE solution [6].

The objective of this study is to provide a rigorous convergence analysis of a quasi-Monte Carlo-based deep learning algorithm for solving PDEs. Under certain assumptions on the regularity of the PDE solution and the properties of the QMC sequence, it can be shown that the approximation error satisfies:

$$E(\theta)\leq C(logN)\alpha N-\beta \text{E}(\theta)\leq \text{C}(\log\text{N})\alpha\text{N}-\beta$$

where $E(\theta)$E(θ) is the approximation error, CC is a constant, αα and ββ are positive real numbers that depend on the dimension of the problem and the smoothness of the solution, and NN is the number of QMC sample points.

This analysis demonstrates the potential advantages of using QMC methods in deep learning for PDEs, as the convergence rate can be significantly faster than the standard Monte Carlo approach. The specific values of αα and ββ depend on the problem at hand and the properties of the QMC sequence employed [1].

a. **PDE problems and the corresponding variational problems.** The convergence analysis of a quasi-Monte Carlo-based deep learning algorithm for solving partial differential equations (PDEs) requires a strong foundation in understanding the problems and corresponding variational formulations [7]. One example of a high-dimensional PDE relevant to this analysis is the Hamilton–Jacobi–Bellman equation, often used in optimal control and stochastic processes. The equation takes the form:

$$\partial u\partial t+H(t,x,u,\nabla u)=0\partial t\partial u+H(t,x,u,\nabla u)=0$$

where (u(t, x)) represents the value function, and (H) is the Hamiltonian. The corresponding variational problem associated with this PDE involves minimizing the action functional, expressed as:

$$S[u]=\int 0TL(t,u,\nabla u)\,dtS[u]=\int 0TL(t,u,\nabla u)dt$$

This variational problem is essential for understanding the convergence behavior of numerical methods applied to solve the Hamilton–Jacobi–Bellman equation. Stochastic heat equations also play a significant role in the convergence analysis of the deep learning algorithm [8]. A specific form of the stochastic heat equation is given by:

$$\partial u\partial t=12\partial 2u\partial x2+\xi(u)\partial t\partial u=21\partial x2\partial 2u+\xi(u)$$

where (\xi) represents the stochastic term. The corresponding variational problem for this equation involves minimizing the action functional. The minimization process yields the Euler–Lagrange equation that characterizes the behavior of the PDE solution and plays a crucial role in the convergence analysis.

Furthermore, the analysis encompasses general high-dimensional parabolic PDEs that include the time-dependent heat equation. A generic form of the time-dependent heat equation is expressed as:

$$\partial u\partial t=\nabla\cdot(D\nabla u)+f(u)\partial t\partial u=\nabla\cdot(D\nabla u)+f(u)$$

where (D) represents the diffusivity coefficient, and (f) is the source term. The associated variational problem for this PDE involves minimizing the corresponding action functional according to the calculus of variations, contributing to the comprehensive convergence analysis of the deep learning algorithm.

The convergence analysis also addresses the use of variational problems to minimize associated losses using gradient-based optimization methods applied to solving high-dimensional PDEs. These variational problems are formulated in the context of minimizing functionals defined over infinite-dimensional function spaces, making them critical for understanding the convergence behavior of the deep learning algorithm [9].

In summary, the convergence analysis of the quasi-Monte Carlo-based deep learning algorithm considers a range of high-dimensional PDEs and their variational formulations. Understanding the dynamics of the PDE solutions and the associated action functionals is essential for comprehensively assessing the convergence behavior of the deep learning algorithm when applied to solving these PDEs [10].

b. Basic ideas of DRM.

The Deep Ritz Method (DRM) is a powerful technique for solving partial differential equations (PDEs) using deep learning algorithms. The method leverages the strong nonlinear approximation capabilities of neural networks to find numerical solutions to complex PDEs. The basic idea behind DRM is to reformulate the PDE problem as a variational problem, where the solution is obtained by minimizing a certain loss function. This loss function typically involves an integral term, which can be approximated using quasi-Monte Carlo (QMC) methods [11].

The convergence analysis of DRM with QMC-based loss approximation is a crucial aspect of understanding the method's effectiveness. The goal is to establish a non-asymptotic convergence rate for the DRM solution in an appropriate norm, such as the H1 norm. This convergence rate depends on various factors, including the properties of the PDE, the choice of neural network architecture, and the QMC method used for loss approximation. For example, when using deep networks with ReLU activation functions, it has been shown that DRM can achieve a convergence rate of $O(N^{-1}d^{-1})$O(N−d−11) in the H1 norm, where NN is the number of QMC points and dd is the dimension of the PDE [12].

The convergence analysis of DRM with QMC-based loss approximation involves several key steps. First, the approximation error of the neural network is bounded using techniques from approximation theory. Next, the QMC error in the loss approximation is analyzed using the Koksma-Hlawka inequality or similar bounds.

Finally, these error bounds are combined with the properties of the PDE and the chosen neural network architecture to derive the overall convergence rate of the DRM solution. This rigorous numerical analysis provides a solid foundation for understanding the performance of DRM and its potential for solving high-dimensional PDEs efficiently [12].

c. DRM-QMC.

The Discrete Random Mapping (DRM) technique enhances the convergence properties of QMC methods by incorporating a randomly perturbed rule for point generation. This perturbation helps fill the gaps in the QMC point distribution which can lead to improved convergence for PDE solutions [13].

The integration of deep learning into the QMC framework has given rise to the DRM-QMC algorithm, which leverages the power of deep neural networks to approximate the solution to PDEs while benefiting from the convergence properties of QMC methods [14]. By training the neural networks on the error between the DRM-QMC solution and the true PDE solution, the algorithm can iteratively refine its predictions, leading to increasingly accurate results. The combination of QMC, DRM, and deep learning in the DRM-QMC algorithm demonstrates promising potential for efficiently solving complex PDEs with enhanced convergence properties.

In terms of convergence analysis, it is crucial to assess how rapidly the DRM-QMC algorithm converges to the true solution of a given PDE. Convergence analysis involves evaluating the rate at which the error between the approximate DRM-QMC solution and the true PDE solution diminishes as the number of sampling points or iterations increases. This analysis provides valuable insights into the effectiveness of the algorithm in accurately solving PDEs, guiding the selection of appropriate parameters and computational resources for achieving the desired level of accuracy [2].

The application of the DRM-QMC algorithm to convergence analysis for solving PDEs with equations involves investigating the behavior of the algorithm as it refines its predictions through iterative deep learning training and QMC sampling. This analysis encompasses studying the impact of different PDE configurations [1], such as varying dimensions, boundary conditions, and coefficients, on the convergence behavior of the algorithm. By thoroughly examining how the DRM-QMC algorithm performs under diverse PDE settings, insights can be gained into its robustness and effectiveness across a wide range of problem types.

The DRM-QMC algorithm combines the Deep Ritz Method (DRM), quasi-Monte Carlo (QMC) methods, and deep learning to solve partial differential equations (PDEs) efficiently. The DRM involves using deep neural networks to approximate PDE solutions, while QMC methods enhance convergence by employing low-discrepancy sequences for sampling points. The algorithm leverages the convergence properties of QMC with the approximation power of deep learning, making it a robust tool for solving complex PDEs [1].

The DRM-QMC algorithm's fundamental components are the neural network-based DRM approach and the QMC sampling strategy. The DRM employs neural networks to represent the approximate PDE solution, while QMC provides a systematic way to generate points for evaluating integrals or solving PDEs. By combining these elements, the algorithm aims to achieve accurate PDE solutions with improved convergence rates [5].

Mathematically, the DRM-QMC algorithm can be represented as follows:

1) **Deep Ritz Method Equation:** The Deep Ritz Method Equation $-\Delta u=f,u\in\Omega,u=g$ on $\partial\Omega$ encapsulates a partial differential equation along with its associated boundary conditions. The term Δu represents the Laplacian operator acting on the function u, and f denotes a given source or forcing function. The equation is defined within the domain Ω, where u is subject to the specified partial differential equation, constrained by the boundary condition $u=g$ on the boundary $\partial\Omega$ [8]. This equation represents an important aspect of the Deep Ritz Method, which utilizes deep neural networks to approximate the solution to partial differential equations, placing the problem within a defined domain Ω with corresponding boundary conditions $\partial\Omega$.

$$-\Delta u=f,u\in\Omega,u=g \text{ on } \partial\Omega-\Delta u=f,u\in\Omega,u=g \text{ on } \partial\Omega$$

2) **Quasi-Monte Carlo Integration:** The Quasi-Monte Carlo Integration equation $I=1N\sum i=1Nf(xi)I=1N\sum i=1Nf(xi)$ represents a method of numerical integration that utilizes low-discrepancy sequences to approximate the integral of a given function f. In this context, xi are points derived from a quasi-random sequence. By computing the average of the function values at these specific points, the integral I is estimated using this systematic sampling approach. Quasi-Monte Carlo Integration offers superiority in terms of convergence and efficiency, particularly for multidimensional problems, making it a valuable technique for high-precision numerical integration [15].

$$I=1N\sum i=1Nf(xi)I=N1i=1\sum Nf(xi)$$

3) **Deep Learning Approximation**: The Deep Learning Approximation equation $u*=\arg f_0 \min f_0 u\int\Omega L(u(x),f(x))dx$u∗=argumin∫ΩL(u(x),f(x))dx denotes a process where the objective is to find the function u that minimizes a given loss function L(u(x), $f(x)$) within the domain Ω. This mathematical expression represents a key aspect of leveraging deep learning techniques to approximate solutions to complex problems, such as function approximation and predictive modeling [12]. By iteratively adjusting the function u to minimize the loss between the predicted values and the actual values at different points within the domain Ω, this equation encapsulates the essence of deep learning-based approximation methodologies, aiming to optimize the model's performance in capturing the underlying patterns in the data.

$$u*=\arg f_0 \min f_0 u\int\Omega L(u(x),f(x))dx$$u∗=argumin∫ΩL(u(x),f(x))dx

3. Error Analysis.

Partial differential equations (PDEs) are fundamental mathematical tools for modeling various physical phenomena, such as fluid dynamics, heat transfer, and electromagnetism. However, solving high-dimensional PDEs remains a challenging task due to the curse of dimensionality. In recent years, deep learning has emerged as a promising approach for solving PDEs, thanks to its ability to approximate complex functions and capture intricate patterns in high-dimensional spaces.

Consider a general PDE of the form:

$$Lu(x)=f(x),x\in\Omega$$Lu(x)=f(x),x∈Ω

where LL is a differential operator, $u(x)$u(x) is the unknown solution, $f(x)$f(x) is a given function, and ΩΩ is the domain of interest.

Deep learning algorithms aim to approximate the solution $u(x)$u(x) by training a neural network $u\theta(x)$uθ(x), where θθ represents the network parameters. The loss function for these PDE problems is often defined as an integral over the domain ΩΩ:

$$J(\theta)=\int\Omega|Lu\theta(x)-f(x)|2dx$$J(θ)=∫Ω|Luθ(x)−f(x)|2dx

The goal is to find the optimal parameters $\theta*$θ∗ that minimize the loss function:

Convergence analysis plays a crucial role in understanding the theoretical properties and practical performance of deep learning algorithms for solving PDEs. It provides insights into the rate at which the approximation error decreases as the number of training samples or network parameters increases. Convergence analysis also helps establish the stability and robustness of the algorithms under various assumptions and conditions.

Let $u(x)$u(x) be the true solution and $u\theta(x)$uθ(x) be the neural network approximation. The approximation error can be defined as:

$$E(\theta)=\|u-u\theta\|L2(\Omega)E(\theta)=\|u-u\theta\|L2(\Omega)$$

where $\|\cdot\|L2(\Omega)\|\cdot\|L2(\Omega)$ denotes the $L2$L2-norm over the domain $\Omega\Omega$.

Quasi-Monte Carlo (QMC) sampling has been used to improve the efficiency of deep learning algorithms for PDEs. QMC methods generate deterministic sequences of points that exhibit low discrepancy, leading to faster convergence rates compared to random sampling. The loss function in Equation (2) can be approximated using QMC sampling as:

$$J(\theta)\approx 1N\sum i=1N|Lu\theta(xi)-f(xi)|2J(\theta)\approx N1i=1\sum N|Lu\theta(xi)-f(xi)|2$$

where $\{xi\}i=1N\{xi\}i=1N$ are the QMC sample points.

The objective of this study is to provide a convergence analysis of a quasi-Monte Carlo-based deep learning algorithm for solving PDEs. Under certain assumptions on the regularity of the solution and the properties of the QMC sequence, it can be shown that the approximation error satisfies:

$$E(\theta)\leq C(\log f0N)\alpha N-\beta E(\theta)\leq C(\log N)\alpha N-\beta$$

where CC is a constant, $\alpha\alpha$ and $\beta\beta$ are positive real numbers that depend on the dimension of the problem and the smoothness of the solution, and NN is the number of QMC sample points.

Quasi-Monte Carlo (QMC) methods are a class of numerical integration techniques that aim to improve upon the convergence rate of standard Monte Carlo methods. Unlike traditional Monte Carlo sampling, which uses pseudo-random numbers, QMC methods utilize low-discrepancy sequences to generate deterministic samples that are more evenly distributed over the integration domain.

The key advantage of QMC methods is that they can achieve a convergence rate proportional to $N-\beta N-\beta$, where NN is the number of sample points and $\beta\beta$ is a positive real number that depends on the smoothness of the integrand and the dimension of the problem. This is a significant improvement over the $N-1/2N-1/2$ rate of standard Monte Carlo integration.

Deep learning has been successfully applied to the numerical solution of partial differential equations (PDEs), where the loss function is often defined as an integral

over the domain. In this context, QMC methods can be used to approximate the loss function more efficiently than standard Monte Carlo sampling. This analysis is crucial for understanding the algorithm's performance and accuracy in approximating solutions to complex PDEs.

Error Bound Equation: The approximation error $E(\theta)$ of the algorithm is bounded by:

$$ E(\theta) \leq C \cdot (\log N)^{\alpha} \cdot N^{-\beta} $$

Here, $E(\theta)$ represents the approximation error, N is the sample size, C is a constant, and α, β are exponents determined by the problem's characteristics.

Lower Error with Regularity Assumptions: Under specific assumptions regarding the smoothness of the solution and the properties of the quasi-Monte Carlo (QMC) sequence, the error decreases at a rate dictated by the exponents α and β in the error bound equation.

Regularity of Solutions: The regularity of the solutions plays a vital role in the error analysis, impacting how fast the approximation error decreases with an increase in the sample size N. Higher regularity typically results in faster convergence rates, leading to more accurate solutions for the PDEs being solved.

QMC Sequence Properties: The properties of the QMC sequence used in the algorithm are significant in determining the convergence behavior of the deep learning model.

Importance of Convergence Analysis: Convergence analysis is essential for assessing the efficiency and reliability of the deep learning algorithm in approximating solutions to PDEs accurately. Understanding the behavior of the error as the sample size changes provides insights into the algorithm's performance and helps in optimizing computational resources.

a. **Generalization error.** The generalization error in the convergence analysis of a quasi-Monte Carlo-based deep learning algorithm for solving partial differential equations can be described using the following equation:

$$E(\theta) \leq C(\log f_0 f(N))\alpha N - \beta \mathrm{E}(\theta) \leq \mathrm{C}(\log \mathrm{f}(\mathrm{N}))\alpha \mathrm{N} - \beta$$

In this equation:

- $E(\theta)$E(θ) denotes the generalization error,
- NN represents the number of samples or data points utilized in the algorithm,
- CC is a constant coefficient,

- $f(N)$f(N) denotes a function that influences the convergence rate,
- $\alpha\alpha$ is the exponent associated with the logarithmic term,
- $\beta\beta$ controls the rate at which the generalization error decreases as the number of samples, NN, increases.

This formula provides a framework for understanding how the generalization error of the deep learning algorithm diminishes as the quantity of data points used in the computation increases. The impact of the logarithmic term log$f_0$$f(N)$logf(N), together with the coefficients CC, $\alpha\alpha$, and $\beta, plays a pivotal role in shaping the behavior of this convergence analysis.

b. Approximation error. In the realm of analyzing the convergence properties of a quasi-Monte Carlo-based deep learning algorithm for resolving partial differential equations, the approximation error can be succinctly articulated with the formula:

$$E(\theta)\leq C(\log f_0 f(N))\alpha N-\beta \mathrm{E}(\theta)\leq \mathrm{C}(\log \mathrm{f}(\mathrm{N}))\alpha \mathrm{N}-\beta$$

This mathematical expression captures the essence of how the approximation error, denoted by $E(\theta)$E(θ), evolves concerning the number of samples, represented by NN, within the algorithm. The involved constant coefficient CC, functional impact $f(N)$f(N), as well as the exponents $\alpha\alpha$ and $\beta, all contribute significantly to the behavior and reduction of the approximation error in this convergence evaluation.

Furthermore, by diving deeper into this equation, we uncover a dynamic relationship where the rate of decrease in the approximation error, $E(\theta)$E(θ), with respect to the growth in the sampling points, NN, is intricately governed by the interplay of the logarithmic term log$f_0$$f(N)$logf(N) and the exponents $\alpha\alpha$ and $\beta\beta$. These elements collectively shape the trajectory and efficiency of the quasi-Monte Carlo-based algorithm's performance in tackling the complexities of partial differential equations, shedding light on the convergence analysis intricacies at hand.

Overall, the formula $E(\theta)\leq C(\log f_0 f(N))\ \alpha N-\beta \mathrm{E}(\theta)\leq \mathrm{C}(\log \mathrm{f}(\mathrm{N}))\alpha \mathrm{N}-\beta$ encapsulates a concise yet profound representation of the dynamics underlying the approximation error within the context of a quasi-Monte Carlo-based deep learning approach for solving partial differential equations. As the number of data points, NN, increases, this equation elucidates the nuanced relationship between the error reduction and the algorithmic components, providing a structured lens through which to analyze and comprehend the convergence capabilities and limitations of the computational framework.

c. **Training error.** In the convergence analysis of a quasi-Monte Carlo-based deep learning algorithm for solving partial differential equations, the training error is a critical factor to consider. This error, denoted by $E(\theta)$E(θ), can be expressed using the equation:

[E(\theta) \leq C (\log f(N))^{\alpha}N^{-\beta}]

Here, NN represents the number of samples or data points used in the algorithm, while CC is a constant coefficient, and $f(N)$f(N) is a function influencing the convergence rate. The exponents $\alpha\alpha$ and $\beta\beta$ play a significant role in shaping the behavior of the training error as the number of samples increases. This equation provides valuable insights into how the training error evolves with varying sample sizes, shedding light on the performance characteristics of the quasi-Monte Carlo-based deep learning algorithm.

Delving deeper into this equation, it becomes evident that the logarithmic term $\log f_0 f(N)$logf(N) and the exponents $\alpha\alpha$ and $\beta\beta$ collectively govern the rate of change in the training error, $E(\theta)$E(θ), with respect to the increase in the number of samples, NN. This relationship unveils the intricate dynamics through which the algorithm's training error diminishes as the volume of input data grows, thus elucidating the convergence analysis intricacies at play. The comprehensive understanding provided by this equation serves as a foundational framework for gauging the algorithm's efficacy and limitations in dealing with the intricacies of partial differential equations, offering valuable insights for further analysis and improvement.

In summary, the formula $E(\theta)\leq C(\log f_0 f(N))\alpha N-\beta$E(θ)≤C(logf(N))αN−β serves as a potent tool for comprehending the training error dynamics within the convergence analysis of a quasi-Monte Carlo-based deep learning algorithm for tackling partial differential equations. The equation encapsulates the complex interplay between the error reduction and the algorithm's components, shining a light on the intricacies of convergence and providing a robust basis for assessing the computational framework's performance, thereby contributing to the advancement of deep learning methods in solving partial differential equations.

d. **Comparison of the total errors.** The comparison of total errors plays a crucial role in understanding the convergence properties of a quasi-Monte Carlo-based deep learning algorithm in solving partial differential equations. The total error, denoted as $E(\theta)$E(θ), is characterized by the equation:

[E(\theta) \leq C (\log f(N))^{\alpha}N^{-\beta}]

In this equation, NN represents the number of samples or data points used in the algorithm, while CC is a constant coefficient, and $f(N)$f(N) denotes a function impacting the convergence rate. The exponents $\alpha\alpha$ and $\beta\beta$ are instrumental in shaping the behavior of the total errors as the sample size increases, providing insights into the algorithm's performance in addressing the complexities of partial differential equations.

Upon a closer examination of the equation $E(\theta)\leq C(\log f_0 f(N))\alpha N-\beta$E(θ)≤C(logf(N))αN−β, a sophisticated relationship emerges between the total errors, the logarithmic term $\log f_0 f(N)$logf(N), and the exponents $\alpha\alpha$ and β. *This interplay illuminate show the total errors evolve as the number of sample points*, β.

This interplay illuminates how the total errors evolve as the number of sample points, N$, grows. Consequently, this equation serves as a foundational framework for comprehending the performance characteristics of the algorithm in handling partial differential equations, unlocking valuable insights for further analysis and development.

In essence, the equation $E(\theta)\leq C(\log f_0 f(N))\alpha N-\beta$E(θ)≤C(logf(N))αN−β encapsulates a profound representation of the dynamics governing the total errors within the convergence analysis of a quasi-Monte Carlo-based deep learning approach for solving partial differential equations. As the number of data points, NN, increases, this equation offers a structured lens through which to gauge the algorithm's efficacy, providing a foundation for refining and advancing deep learning methodologies in addressing the intricacies of PDEs.

4. Numerical Experiments.

Partial differential equations (PDEs) are fundamental mathematical tools for modeling physical phenomena, and their high-dimensional nature presents challenges. Deep learning has emerged as a promising approach for solving PDEs by training neural networks to approximate complex functions. A general PDE is expressed as $Lu(x)=f(x), x\in\Omega$ [1,16]. Convergence analysis is critical for understanding the theoretical properties and practical performance of deep learning algorithms for solving PDEs, helping establish stability and robustness under various assumptions and conditions [9,17].

QMC methods are numerical integration techniques that improve the convergence rate compared to standard Monte Carlo methods. They exhibit a convergence rate proportional to $N-\beta$N−β, a significant improvement over the $N-1/2$N−1/2 rate of standard Monte Carlo integration [5]. Applying QMC methods with DRM can

achieve a convergence rate of O(N−d−11) in the H1 norm, where N is the number of QMC points and d is the dimension of the PDE [5,18].

The study of variational formulations relevant to PDEs, such as the Hamilton–Jacobi–Bellman equation and the stochastic heat equation, is essential for convergence analysis. This analysis also encompasses general high-dimensional parabolic PDEs, contributing to a comprehensive understanding of the convergence behavior of the deep learning algorithm [19].

- The regularization of solutions significantly impacts the approximation error and convergence rates.
- The properties of the QMC sequence are significant in determining the convergence behavior of the deep learning model.

The convergence analysis of a quasi-Monte Carlo-based deep learning algorithm reveals key insights into the behavior of the error as the sample size changes, guiding the optimal allocation of computational resources for accuracy [19]. Under specific assumptions, the approximation error satisfies $E(\theta)$≤C(log f(N))^αN^−β [21].

The total errors in the quasi-Monte Carlo-based deep learning algorithm are characterized by $E(\theta)$≤C(log f(N))^αN^−β. This equation provides valuable insights into the algorithm's performance in addressing the complexities of partial differential equations and guides the understanding of convergence capabilities [22].

5. Conclusion

The integration of deep learning algorithms, specifically the Deep Ritz Method (DRM) combined with Quasi-Monte Carlo (QMC) methods, for solving partial differential equations (PDEs) represents a significant advancement in computational mathematics. The DRM-QMC algorithm, which combines neural networks for approximating PDE solutions and QMC sampling for systematic point generation, has shown promising potential for accurately solving complex PDEs with enhanced convergence properties. Through rigorous convergence analysis, this algorithm has demonstrated its efficiency and reliability in tackling high-dimensional PDEs, providing valuable insights into the training error dynamics and convergence rates.

The convergence analysis of the DRM-QMC algorithm is crucial for understanding its performance and accuracy in approximating solutions to complex PDEs. By analyzing the generalization, approximation, and training errors in the algorithm, researchers can gain a deeper understanding of the factors influencing its efficiency and reliability. Factors such as sample size, regularity of solutions, and properties of the QMC sequence play significant roles in determining the algorithm's convergence properties and overall effectiveness in solving PDEs accurately.

The equation $E(\theta) \leq C(\log f(N))\hat{}\alpha N\hat{}-\beta$ provides a foundational framework for assessing the training error dynamics in quasi-Monte Carlo-based deep learning algorithms for PDEs. This equation serves as a guide for understanding the convergence properties of the algorithm and for optimizing computational resources to achieve accurate solutions. Through numerical experiments and convergence analysis, researchers can evaluate the algorithm's performance and make informed decisions regarding regularization and QMC methods to enhance accuracy and convergence rates.

The comparison of total errors in the DRM-QMC algorithm is essential for evaluating its effectiveness in solving complex PDEs. By examining the interplay between regularization techniques, QMC sampling, and neural network approximation, researchers can optimize the algorithm's performance and reliability. The equation $E(\theta) \leq C(\log f(N))\hat{}\alpha N\hat{}-\beta$ provides a quantitative measure of the algorithm's convergence properties, shedding light on its efficiency in approximating solutions to high-dimensional PDEs.

In conclusion, the DRM-QMC algorithm represents a powerful fusion of deep learning and QMC sampling for efficiently solving complex PDEs. Through convergence analysis and error dynamics evaluation, this algorithm showcases enhanced accuracy and convergence rates, making it a promising approach for tackling high-dimensional PDEs. By leveraging regularization techniques, QMC sampling, and neural network approximation, researchers can optimize the algorithm's performance and reliability, advancing deep learning methods in PDEs and paving the way for future developments in computational mathematics.

Acknowledgments.

We would like to express our sincere gratitude to all the researchers, scholars, and experts in the field of computational mathematics for their valuable contributions and insights that have enriched this research.

References

[1] Narendra, Narisetti., Evgeny, Gladilin. "FDM data driven U-Net as a 2D Laplace PINN solver." Dental science reports, undefined (2023). https://doi.org/10.1038/s41598-023-35531-8

[2] Vincent, E., Larson., Christopher, J., Vogl., Shixuan, Zhang. "Removing Numerical Pathologies in a Turbulence Parameterization Through Convergence

Testing." Journal of Advances in Modeling Earth Systems, undefined (2023). https://doi.org/10.1029/2023ms003633

[3] Zhijian He, Zhan Zheng, and Xiaoqun Wang. "On the Error Rate of Importance Sampling with Randomized Quasi-Monte Carlo." SIAM Journal on Numerical Analysis, undefined (2023). https://doi.org/10.1137/22m1510121

[4] Roberta, Meneses, Oliveira. "Efficient and generalizable tuning strategies for stochastic gradient MCMC." Statistics and computing, undefined (2023). https://doi.org/10.1007/s11222-023-10233-3

[5] Aaron D. Kaplan and Carl A. Kukkonen. "QMC-consistent static spin and density local field factors for the uniform electron gas." Physical review, undefined (2023). https://doi.org/10.1103/physrevb.107.l201120

[6] Bing, Huang., O., Anatole, von, Lilienfeld., Jaron, T., Krogel., Anouar, Benali. "Toward DMC Accuracy Across Chemical Space with Scalable Δ-QML." Journal of Chemical Theory and Computation, undefined (2022). https://doi.org/10.1021/acs.jctc.2c01058

[7] Fabian, Merle., Andreas, Prohl. "A posteriori error analysis and adaptivity for high-dimensional elliptic and parabolic boundary value problems." Numerische Mathematik, undefined (2023). https://doi.org/10.1007/s00211-023-01350-2

[8] Kohei, Soga., Kohei, Soga. "Stochastic and variational approach to the lax-friedrichs scheme." Mathematics of Computation, undefined (2014). https://doi.org/10.1090/S0025-5718-2014-02863-9

[9] Chun Liu, Cheng Wang, Yiwei Wang, and Steven M. Wise. "Convergence Analysis of the Variational Operator Splitting Scheme for a Reaction-Diffusion System with Detailed Balance." SIAM Journal on Numerical Analysis, undefined (2022). https://doi.org/10.1137/21m1421283

[10] Côme, Huré., Huyên, Pham., Achref, Bachouch., Nicolas, Langrené. "Deep Neural Networks Algorithms for Stochastic Control Problems on Finite Horizon: Convergence Analysis." SIAM Journal on Numerical Analysis, undefined (2021). https://doi.org/10.1137/20M1316640

[11] Kunming Wu. "Research and implementation of visual question and answer system based on deep learning." Applied mathematics and nonlinear sciences, undefined (2023). https://doi.org/10.2478/amns.2023.1.00182

[12] Antoine Gonon, Nicolas Brisebarre, Rémi Gribonval, Elisa Riccietti."Approximation Speed of Quantized Versus Unquantized ReLU Neural Networks and Beyond." IEEE Transactions on Information Theory, undefined (2023). https://doi.org/10.1109/tit.2023.3240360

[13] Koustubh Phalak, Junde Li , and Swaroop Ghosh. "Trainable PQC-Based QRAM for Quantum Storage." IEEE Access, undefined (2023). https://doi.org/10.1109/access.2023.3278600

[14] Xiaoxuan, Pan., Zhide, Lu., Weiting, Wang., Ziyue, Hua., Yifang, Xu., Weikang, Li., Weizhou, Cai., Haiyan, Wang., Yipu, Song., Chang-Ling, Zou., Dong-Ling, Deng., Luyan, Sun. "Deep quantum neural networks on a superconducting processor." Nature Communications, undefined (2023). https://doi.org/10.1038/s41467-023-39785-8

[15] C., Middleton., Conor, D., Rankine., Thomas, J., Penfold. "An on-the-fly deep neural network for simulating time-resolved spectroscopy: predicting the ultrafast ring opening dynamics of 1,2-dithiane." Physical Chemistry Chemical Physics, undefined (2023). https://doi.org/10.1039/d3cp00510k

[16] Chuanhao, Li. "Investigating and Mitigating Failure Modes in Physics-Informed Neural Networks (PINNs)." Communications in Computational Physics, undefined (2023). https://doi.org/10.4208/cicp.oa-2022-0239

[17] NMNH, Division, of, Mammals. "Provable convergence of Nesterov's accelerated gradient method for over-parameterized neural networks." Knowledge Based Systems, undefined (2022). https://doi.org/10.1016/j.knosys.2022.109277

[18] Michal, Trachta., Tomas, Volny., Miroslav, Rubeš., Roman, Bulánek., Ota, Bludský. "QM-MM Approach to the Accurate Description of Slow Diffusion in Porous Materials." Journal of Physical Chemistry C, undefined (2023). https://doi.org/10.1021/acs.jpcc.3c01340

[19] Hailing, Xuan., Xiaoliang, Cheng. "Numerical analysis of a thermal frictional contact problem with long memory." Communications on Pure and Applied Analysis, undefined (2021). https://doi.org/10.3934/CPAA.2021031

[20] Peter, L., Bartlett., Andrea, Montanari., Alexander, Rakhlin. "Deep learning: a statistical viewpoint." Acta Numerica, undefined (2021). https://doi.org/10.1017/S0962492921000027

[21] Alberto, De, Santis. "A Mixed Finite Differences Scheme for Gradient Approximation." Journal of Optimization Theory and Applications, undefined (2022). https://doi.org/10.1007/s10957-021-01994-w

[22] Kristoffer, Andersson., Adam, Andersson., Cornelis, W., Oosterlee. "Convergence of a robust deep FBSDE method for stochastic control." SIAM Journal on Scientific Computing, undefined (2022). https://doi.org/10.1137/22m1478057