

Hybrid Quantum-Classical Computing for Physical Problems: Architectures, Algorithms, and Applications in the Networked Era

Baixin Pan

The University of Hong Kong, Hong Kong, China

Keywords: Hybrid Quantum-Classical Computing, Physical Problems, Quantum Algorithms, High-Performance Computing, Quantum Error Correction, Computational Physics, Materials Science, Fluid Dynamics, High-Energy Physics, Optimization

Abstract: This report provides a comprehensive analysis of hybrid quantum-classical computing's role in addressing complex physical problems. It delineates the foundational principles of both classical and quantum paradigms, explores advanced architectural models and integration strategies, and details the application of cutting-edge quantum algorithms—including Variational Quantum Eigensolver (VQE), Quantum Phase Estimation (QPE), and Quantum Approximate Optimization Algorithm (QAOA)—alongside classical numerical methods such as Finite Element Method (FEM) and Finite Difference Method (FDM). Through a review of real-world case studies in materials science, chemistry, fluid dynamics, and high-energy physics, the report illustrates the transformative potential of this synergistic approach. Furthermore, it critically examines the prevailing challenges, encompassing hardware limitations, quantum error correction, software-hardware co-design, scalability, and data handling complexities, while forecasting the convergence with exascale computing. The aim is to elucidate the current state and future trajectory of hybrid quantum-classical computing as a pivotal tool for scientific discovery and engineering innovation in the networked era.

1. Introduction

1.1 Overview of the Computational Landscape: Classical vs. Quantum

Classical computing, which forms the bedrock of contemporary digital infrastructure, operates on a binary system where data is represented by bits (0s or 1s) and processed sequentially. While immensely powerful for a vast array of tasks, this approach encounters significant limitations when confronted with problems that demand extensive computational resources, particularly those exhibiting exponential scaling of complexity {AroraBarak2009}[1]. Modern devices, with processing power over 100 times that of machines from the 1990s, owe their rapid evolution to the exponential increase in transistors and their computational abilities {Moore1965}[2]. However, as silicon chip and superconductor development approaches material limits, a path forward is sought for certain computationally intensive problems {Shalf2020}[3]. Classical systems, despite their advancements,

inherently struggle with NP-hard and NP-complete problems, where the solution time scales exponentially with problem size, demanding prohibitive computational resources {GareyJohnson1979}[4].

In contrast, quantum computing harnesses the unique principles of quantum mechanics, such as superposition, entanglement, and interference {NielsenChuang2010}[5]. Qubits, unlike classical bits, can exist in a superposition of multiple states concurrently, meaning they can represent both 0 and 1 simultaneously. This inherent parallelism enables quantum computers to process millions of operations simultaneously, leading to exponentially faster computations for certain classes of problems {Deutsch1992}[6]. Qubits can be realized through various physical systems, such as the spin-up or spin-down motion of an electron, the polarization of a photon, or the ground and excited energy states of a particle, all representing a two-level quantum system {Ladd2010}[7]. Through entanglement, their states become interconnected regardless of their physical separation, allowing quantum processors to infer information about one particle by measuring another. The universe itself is fundamentally quantum mechanical, characterized by inherent uncertainty and probabilistic behaviors, making quantum computers uniquely designed to model and simulate these complex quantum phenomena {Feynman1982}[8].

1.2 Motivation for Hybrid Quantum-Classical Approaches in Tackling Intractable Physical Problems

The current state of quantum hardware is characterized by the Noisy Intermediate-Scale Quantum (NISQ) era {Preskill2018}[9]. Devices in this era possess a limited number of qubits, restricted connectivity, and are highly susceptible to noise and decoherence {Bharti2022}[10]. These limitations currently prevent the full realization of quantum advantage with purely quantum algorithms for many complex problems. The inherent fragility of qubits, sensitive to environmental factors like heat, radiation, and thermal fluctuations, leads to rapid loss of their quantum state (decoherence) and high error rates. The need for fault-tolerant quantum computing (FTQC) to achieve industrially-relevant applications is widely acknowledged, requiring thousands or even millions of physical qubits {Fowler2012}[11].

Hybrid quantum-classical (HQC) computing emerges as a pragmatic solution, synergistically combining the strengths of both paradigms {Moll2018}[12]. Quantum processors are delegated computationally intensive tasks where they can offer a unique advantage, such as optimization, complex data encoding, and quantum simulations {Peruzzo2014}[13]. Classical systems, meanwhile, provide the robust, well-established infrastructure for large-scale data preprocessing, neural network training, output interpretation, error mitigation, and overall control {Endo2021}[14]. This integrated approach aims to circumvent current hardware limitations and accelerate the path toward practical quantum advantage.

Many critical physical problems, such as simulating intricate molecular interactions, accelerating drug discovery, designing novel materials, and solving large-scale optimization challenges, are currently intractable for classical computers due to the exponential scaling of their computational requirements {Bauer2020}[15]. HQC approaches offer a promising avenue to address these formidable challenges, unlocking new frontiers in scientific research and engineering innovation.

The current landscape of quantum computing development indicates a critical, near-term dependency: the practical path to demonstrating quantum advantage, or "quantum utility," is not through purely quantum systems in the noisy NISQ era, but specifically through intelligent integration within hybrid architectures. This observation arises from the consistent emphasis on the limitations of current quantum hardware, such as noise, limited qubit counts, and decoherence, alongside the frequent presentation of hybrid approaches as the immediate and promising solution for achieving

computational utility. This implies a significant shift in the strategic focus of quantum computing research and development. The immediate goal is less about achieving a definitive "quantum supremacy," where quantum computers solve problems provably intractable for any classical computer, and more about demonstrating "quantum utility," where quantum computers provide a practical, even if not exponentially faster, advantage for specific sub-problems when integrated within classical workflows [Lubinski2021][16]. This pragmatic shift from "supremacy" to "utility" is crucial for fostering commercialization and widespread adoption, as it allows for immediate, tangible benefits despite the inherent imperfections of current quantum hardware. Furthermore, it underscores that the development of robust classical-quantum interfaces, sophisticated workflow management tools, and efficient communication protocols are as critical to progress as advancements in qubit coherence and count.

Discussions of "quantum advantage" have traditionally centered on the exponential speedup offered by algorithms like Shor's [Shor1994][17]. However, a more nuanced perspective is emerging, where "quantum advantage" in the hybrid era may not always manifest as an exponential speedup over all classical methods for the entire problem. Instead, it can represent a significant practical improvement in specific, computationally intensive subroutines or for problems where classical brute-force methods are the only viable, albeit time-consuming, alternatives. For instance, a recent study demonstrated that a quantum algorithm achieved comparable or better solutions for hard optimization problems in seconds, whereas classical methods required tens of seconds or more, even with classical preprocessing and postprocessing [Harrigan2021][18]. This redefines the immediate narrative surrounding quantum computing's impact. Rather than waiting for a fully fault-tolerant universal quantum computer to solve all problems, the focus shifts to identifying and accelerating "quantum-suitable" components within existing classical workflows. This pragmatic view encourages industry adoption by demonstrating incremental, yet valuable, performance gains for real-world problems, even if the ultimate goal of full exponential speedup for arbitrary problems remains a long-term endeavor. This also highlights the increasing importance of sophisticated problem decomposition techniques and the precise identification of sub-problems where quantum acceleration can provide a measurable and practical benefit.

2. Foundational Computational Paradigms

2.1 Principles of Quantum Computing

Quantum computing is a multidisciplinary field that integrates aspects of computer science, physics, and mathematics, leveraging quantum mechanical effects such as superposition and quantum interference to solve problems more efficiently than classical computers. The field encompasses both hardware research and application development.

Qubits: Unlike classical bits that are limited to discrete states of 0 or 1, quantum bits (qubits) are represented by quantum particles and can exist in a superposition of multiple states simultaneously. This inherent parallelism is a fundamental source of quantum computers' processing power, enabling them to evaluate millions of possibilities concurrently. Qubits can be realized through various physical systems, such as the spin-up or spin-down motion of an electron, the polarization of a photon, or the ground and excited energy states of a particle, all representing a two-level quantum system. The value of a qubit remains unknown until it is measured, at which point it resolves into a deterministic state of either $|0\rangle$ or $|1\rangle$.

Entanglement: This unique quantum property occurs when two or more qubits become intrinsically linked, such that the state of one qubit instantaneously influences the state of another, regardless of their physical separation. Entanglement significantly amplifies the power of parallel computation, allowing quantum processors to infer information about one particle by measuring another. When a

quantum state is measured, its wavefunction collapses, and the qubit resolves into a deterministic state (0 or 1), acting as a classical bit.

Decoherence: A critical challenge in quantum computing is decoherence, which refers to the loss of a qubit's fragile quantum state due to undesirable interactions with its environment, such as radiation or thermal fluctuations{Zurek2003}[19]. A major engineering hurdle in constructing quantum computers involves designing specialized structures and control mechanisms to shield qubits from external fields and delay decoherence, thereby preserving the quantum state for longer durations.

Quantum Hardware Components: A quantum computer typically comprises three main components:

1) **Quantum data plane:** This is the core of the quantum computer, containing the physical qubits and the structures required to hold them in place.

2) **Control and measurement plane:** This component converts digital signals into analog or wave control signals, which are then used to perform operations on the qubits in the quantum data plane.

3) **Control processor plane and host processor:** This part implements the quantum algorithm or sequence of operations. The host processor interacts with the quantum software and provides a digital signal or classical bit sequence to the control and measurement plane.

Quantum Software: This layer is responsible for implementing unique quantum algorithms using quantum circuits, which are computing routines that define a series of logical quantum operations on the underlying qubits. Developers utilize various software development tools and libraries to code these algorithms{LaRose2019}[20].

2.2 High-Performance Computing (HPC)

High-Performance Computing involves the coordinated operation of multiple interconnected robust computers, typically organized into a cluster, to process and analyze vast amounts of data and execute complex calculations at high speeds. HPC systems can be deployed on-premises, in the cloud, or in hybrid configurations.

Core Components: HPC architecture is fundamentally built upon three main components :

1) **Compute:** Individual computers or "nodes" within a cluster are dedicated to processing data, executing software or algorithms, and solving problems. Algorithms and software are often run in parallel on each node, with each node responsible for a different task.

2) **Storage:** High-performance storage solutions are essential for managing and retrieving the large datasets generated and processed by the computing component. This can include traditional physical storage, cloud storage, or hybrid solutions.

3) **Network:** High-bandwidth, low-latency interconnects enable rapid communication and data exchange among nodes and storage systems. Fast data transfer speeds are crucial for HPC projects, which often involve nodes exchanging computation results. Specialized hardware accelerators such as Graphics Processing Units (GPUs) and Field-Programmable Gate Arrays (FPGAs) are frequently integrated to significantly enhance computational efficiency.

Capabilities and Use Cases: HPC is a critical enabler for computationally demanding tasks across diverse sectors, including academic and scientific research, engineering design and simulation, and business intelligence. Its applications span from analyzing astronomical data and modeling complex physical scenarios (e.g., automobile collisions, airflow over airplane wings, drug-cell interactions) to designing new machines, optimizing large datasets (e.g., financial portfolios, logistics), and performing large-scale forecasting (e.g., weather prediction, climate change modeling). HPC also provides the foundation for much of the innovation seen in AI and machine learning by supporting complex calculations.

Benefits: HPC offers substantial advantages, including superior speed and performance, reducing

tasks that could take weeks or months on regular computing systems to hours. Cloud-based HPC provides enhanced flexibility and efficiency, allowing workloads to scale up or down as needed and be accessed globally. These efficiencies translate into significant cost savings. Furthermore, HPC systems exhibit inherent fault tolerance, meaning that if one node in a cluster fails, the rest of the system remains resilient.

Limitations: Despite its power, HPC deployment and maintenance require colossal investments in infrastructure, ongoing upkeep, and specialized expertise to ensure smooth operations. Furthermore, classical HPC systems inherently struggle with NP-hard and NP-complete problems, where the solution time scales exponentially with problem size, demanding prohibitive computational resources.

2.3 Classical Numerical Methods for Physical Problems: Finite Element Method (FEM) and Finite Difference Method (FDM)

Numerical methods are indispensable for solving differential equations that describe the underlying physics in many scientific and engineering models. Among these, the Finite Element Method (FEM) and Finite Difference Method (FDM) are two of the most commonly used approaches.

Finite Element Method (FEM): FEM is a widely adopted numerical method for solving differential equations that arise in engineering and mathematical modeling, particularly effective for boundary value problems. It operates by subdividing a complex system or domain into smaller, simpler parts called finite elements {Zienkiewicz2005} [21]. The simple equations modeling these elements are then assembled into a larger system of algebraic equations that represent the entire problem.

A primary advantage of FEM is its exceptional ability to handle complicated geometries, multi-material systems, and non-linearities with relative ease. It allows for flexible mesh adaptivity and provides piecewise approximations, ensuring continuity along element boundaries. FEM can also evaluate values between nodes through interpolation using shape functions. The application of FEM typically involves two essential steps: first, rephrasing the original boundary value problem into its "weak form" (often a manual, theoretical step), followed by the "discretization" of the weak form into a finite-dimensional space. FEM is extensively used in fields such as structural analysis, heat transfer, fluid flow, mass transport, and electromagnetic potential, especially for analyzing problems over complicated or changing domains.

Finite Difference Method (FDM): FDM constitutes a class of numerical techniques for solving differential equations by approximating derivatives with finite differences {LeVeque2007} [22]. Both the spatial and, if applicable, time domains are discretized into a finite number of intervals, and the solution values at these discrete points are approximated by solving algebraic equations derived from finite differences and values from neighboring points.

FDM is generally straightforward to implement. It makes point-wise approximations and is typically restricted to handling rectangular or block-shaped models and simple alterations thereof, making it less flexible for irregular CAD geometries compared to FEM. To apply FDM, the problem's domain must first be discretized, usually into a uniform grid. This process converts ordinary or partial differential equations into a system of linear equations solvable by matrix algebra techniques. The two primary sources of error in FDM are round-off error (due to computer rounding of decimal quantities) and truncation or discretization error (the difference between the exact solution of the original differential equation and the exact quantity assuming perfect computation).

Relationship between FEM and FDM: While distinct, FDM can be considered a particular case of the FEM approach in certain ways; for example, first-order FEM is identical to FDM for Poisson's equation if the problem is discretized by a regular rectangular mesh with each rectangle divided into two triangles. Generally, FEM often provides higher accuracy with fewer elements and superior capability in modeling curved boundaries and nonlinear problems compared to FDM. Both methods

divide the domain into a grid of nodes to obtain a system of equations that can be solved numerically to approximate the solution.

The fundamental advantage of quantum computing stems from superposition and entanglement, enabling millions of operations simultaneously and the exploration of vast solution spaces. Conversely, classical computing, particularly through HPC, excels at processing vast amounts of data and executing complex simulations at remarkable speed and efficiency via highly optimized sequential and parallel processing across multiple nodes. This is not merely a competition of "which is better," but rather a profound synergy. Quantum systems are uniquely suited for modeling inherently quantum phenomena or for exploring combinatorial possibilities that overwhelm classical brute force. Classical systems, however, remain unparalleled for deterministic, high-throughput data management, precise control, and the application of established numerical methods for well-defined problems. This reinforces the fundamental necessity of the hybrid paradigm. Quantum computers are not poised to become general-purpose replacements for classical ones. Instead, they are evolving as specialized accelerators, designed to tackle specific, intractable sub-problems within larger, predominantly classical workflows. This division of labor suggests that the most impactful advancements will arise from the seamless integration and optimization of these complementary computational strengths.

A recurring theme across the available information is the consistent linkage of both quantum and classical computing advancements to the imperative of solving "complex problems" and "physical problems". Whether the objective is to simulate intricate molecular interactions, optimize global logistics networks, or analyze fundamental particle physics data, the underlying motivation for the development of both HPC and quantum technologies is to push the boundaries of what can be computationally modeled, understood, or solved in the physical world. The inherent limitations of classical numerical methods like FEM and FDM when faced with highly complex geometries, multi-material systems, or severe non-linearities directly underscore the compelling need for and motivate the exploration of quantum enhancements. This highlights that the very nature of "physical problems" serves as the primary, overarching driver for innovation in both classical and quantum computing paradigms. The relentless pursuit of more accurate, efficient, and comprehensive simulations or faster, more optimal solutions to intractable optimization problems is what continuously pushes the boundaries of computational capabilities. Hybrid computing emerges as the most viable and effective near-term strategy precisely because it directly addresses the current computational bottlenecks in these critical physical domains by intelligently combining the best available tools from both classical and quantum realms. This strongly implies that sustained, deep interdisciplinary collaboration among physicists, chemists, engineers, and computer scientists will be absolutely paramount for realizing the full, transformative potential of hybrid systems.

3. Hybrid Quantum-Classical Architectures and Integration Strategies

3.1 Architectural Models and Integration Strategies

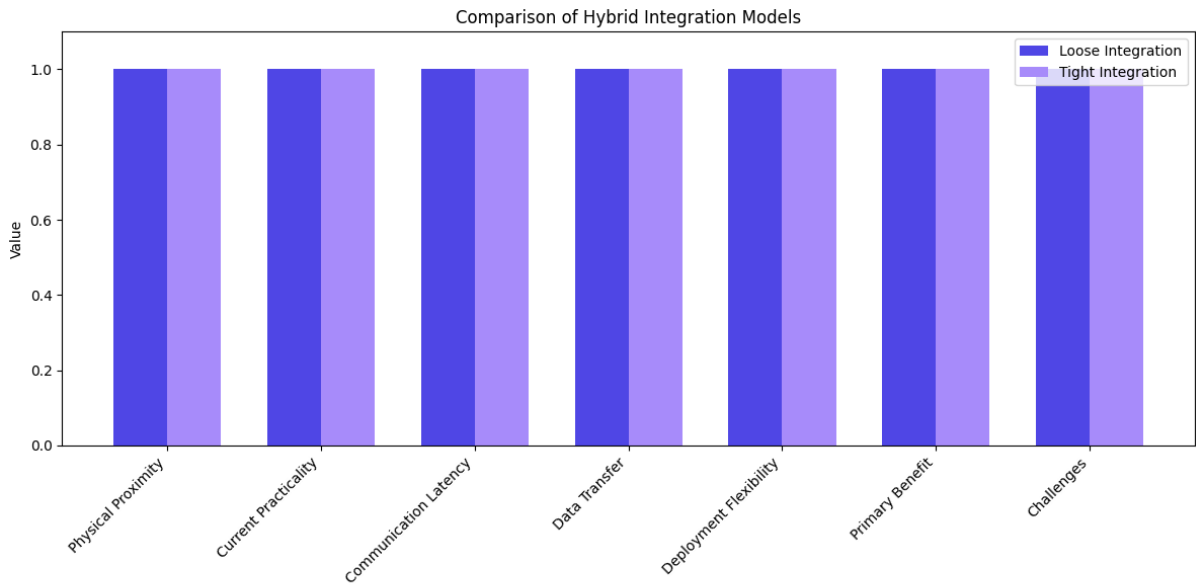
Hybrid quantum-classical (HQC) systems are designed to leverage the unique strengths of quantum computing for specific computationally intensive tasks, such as optimization, pattern recognition, and feature extraction, while relying on classical systems for their established efficiency and maturity in other areas, including large-scale data preprocessing, neural network training, and output interpretation.

- **Loose Integration:** In this model, the Quantum Processing Unit (QPU) is physically separated from the High-Performance Computing (HPC) system, with connectivity established via a network, whether on-premises or cloud-based [McCaskey2020][23]. This architectural approach is currently more practical and is expected to remain prevalent in the near future due to its flexibility and ease of

deployment. However, it introduces crucial research challenges related to network latency and data transfer.

- **Tight Integration:** This model involves the QPU being located directly on an HPC node, aiming for a very close coupling between quantum and classical resources. The primary motivation for tight integration is to minimize communication latency and maximize data transfer bandwidth, thereby optimizing performance for highly interactive hybrid algorithms{Fu2019}[24].

The bar chart, "*Figure 1: Comparison of Hybrid Integration Models (Loose vs. Tight)*," visually contrasts key attributes of these models. Each attribute—Physical Proximity, Current Practicality, Communication Latency, Data Transfer, Deployment Flexibility, Primary Benefit, and Challenges—is evaluated qualitatively. A value of 1 signifies "more practical" or "higher latency," while 0 indicates "less practical" or "lower latency," as noted. The consistent height of the bars at 1 across all categories for both integration types suggests a simplified, illustrative comparison rather than a nuanced quantitative analysis, aiming to highlight these fundamental conceptual differences.



*Note: Values 0 and 1 represent qualitative comparisons for visualization purposes (e.g., 1 for 'more practical' or 'higher latency', 0 for 'less practical' or 'lower latency').

Figure 1: Comparison of Hybrid Integration Models (Loose vs. Tight)

- **Middleware:** Advanced middleware solutions are indispensable for seamlessly integrating quantum circuits with classical computing resources. These middleware layers facilitate efficient interaction between QPUs and HPC compute clusters, regardless of whether they are co-located or distributed across a cloud environment. Pilot-Quantum, for instance, demonstrates how middleware can manage heterogeneous resources, workloads, and tasks through a sophisticated hierarchy of abstraction layers: the Resource Layer (L1) manages and abstracts classical and quantum resources; the Task Layer (L2) focuses on operational management of task execution on a single resource; the Workload Layer (L3) oversees interdependent quantum and classical tasks; and the Workflow Layer (L4) encapsulates the logical dependencies and orchestration of tasks.

Integration Paradigms (Pilot-Quantum): Middleware frameworks often support various integration paradigms to optimize different types of hybrid workflows:

- **HPC-for-Quantum Mode:** This mode focuses on leveraging HPC resources to accelerate and optimize complex quantum algorithms that require strong coupling with classical processes, such as the classical optimization loop in variational quantum algorithms.

- **Quantum-in-HPC Mode:** This paradigm centers on integrating loosely-coupled quantum processes to enhance existing HPC capabilities, where quantum computations act as specialized

accelerators for specific subroutines.

- **Quantum-about-HPC Mode:** This approach incorporates quantum processes into broader classical workflows, often for tasks like data preprocessing, feature extraction, or results analysis, where quantum insights can inform classical computations.

The evolution of quantum computing has seen a critical shift in focus from merely increasing raw qubit count and improving individual qubit coherence times to addressing the inter-component communication bottleneck within hybrid systems. While quantum processors are designed for specific, computationally intensive tasks, their effective integration with classical HPC systems presents significant technical challenges, particularly concerning efficient data movement and job scheduling. The primary challenge is not solely the number of qubits, but how to network many QPUs into a larger, cohesive system, requiring tight integration with HPC. This observation highlights that the scaling of quantum computers for practical applications is no longer solely a matter of quantum hardware development; it is equally, if not more, about the systems integration and software layers that enable seamless communication and workload distribution between quantum and classical components. The emphasis on middleware, flexible programming interfaces, and unified resource management systems underscores the recognition that the "friction" between quantum and classical operations, particularly data transfer bottlenecks and latency, can severely impede the overall performance and utility of hybrid systems{Mccaskey2022}[25]. This indicates that future advancements will heavily rely on innovations in classical-quantum interfaces and distributed computing paradigms, rather than just isolated improvements in quantum hardware.

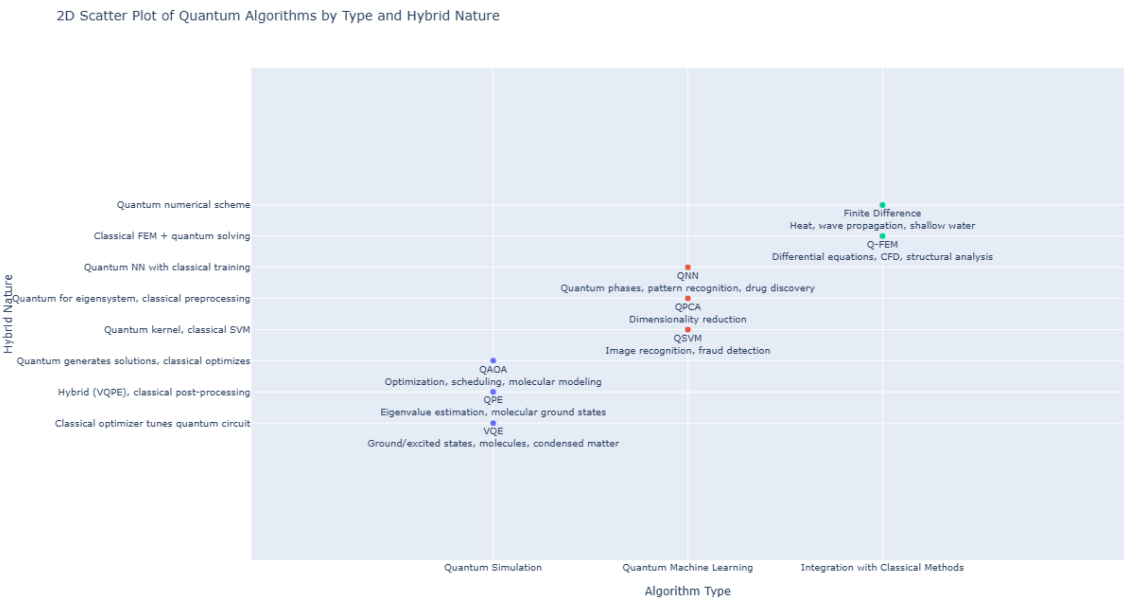
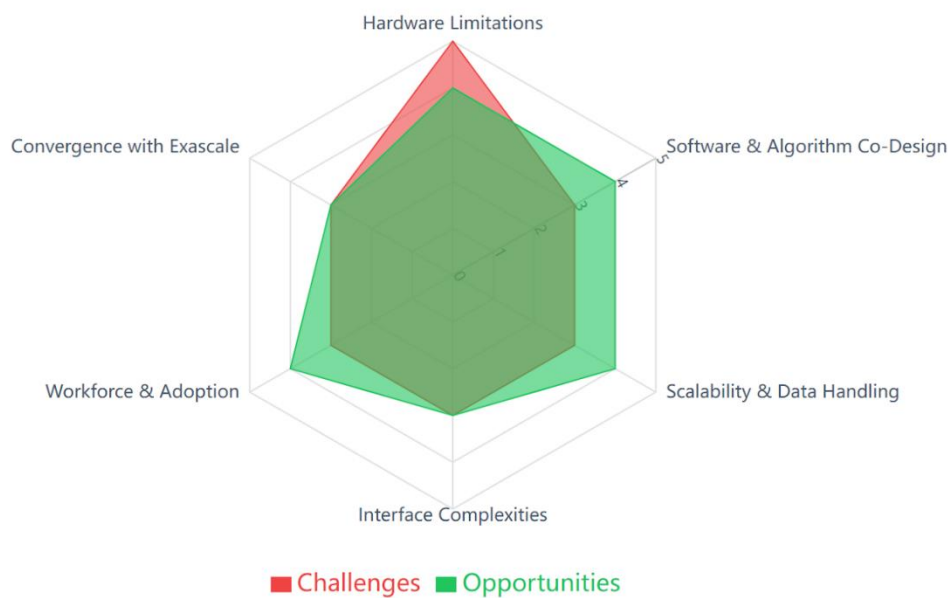


Figure 2: Key Quantum Algorithms and Their Applications in Physical Problems

The 2D scatter plot, "*Figure 2: Key Quantum Algorithms and Their Applications in Physical Problems*," categorizes various quantum algorithms based on their "Algorithm Type" (Quantum Simulation, Quantum Machine Learning, Integration with Classical Methods) and "Hybrid Nature." The plot illustrates how different algorithms, such as Quantum numerical schemes, Quantum NN with classical training, and Hybrid (VQE) with classical post-processing, are positioned across these dimensions. Specific applications like optimization, image recognition, and differential equations are associated with these algorithms, providing a high-level overview of the landscape of quantum computing applications and the role of hybrid approaches. This complex interplay is further

contextualized in the radar chart, "Figure 3: Challenges and Opportunities in Hybrid Quantum-Classical Computing," which qualitatively assesses the landscape of hybrid quantum computing. It presents a comparative view of challenges (red) and opportunities (green) across six dimensions: Hardware Limitations, Software & Algorithm Co-Design, Scalability & Data Handling, Interface Complexities, Workforce & Adoption, and Convergence with Exascale. Each dimension is scored from 1 to 5, where higher values denote greater presence or impact. Notably, "Hardware Limitations" appears as a significant challenge, while "Software & Algorithm Co-Design" and "Scalability & Data Handling" present considerable opportunities, illustrating key areas for focus in the field.

Challenges vs. Opportunities in Hybrid QC



*Note: Values represent a qualitative score (1-5) for challenges and opportunities within each category. Higher values indicate greater presence/impact.

Figure 3: Challenges and Opportunities in Hybrid Quantum-Classical Computing

4. Conclusion

Hybrid quantum-classical computing represents a pivotal evolution in the computational landscape, offering a pragmatic pathway to address complex physical problems that remain intractable for purely classical systems. The analysis presented herein underscores that the current era, characterized by Noisy Intermediate-Scale Quantum (NISQ) devices, necessitates this synergistic approach. Quantum processors, with their unique capabilities in superposition and entanglement, are emerging as specialized accelerators for computationally intensive sub-problems, while robust High-Performance Computing (HPC) systems continue to manage the vast majority of data processing, control, and established numerical methods.

The consistent drive to solve complex physical problems, from molecular simulations in chemistry and materials science to turbulence modeling in fluid dynamics and fundamental calculations in high-energy physics, serves as the primary impetus for innovation in both classical and quantum computing. The limitations of traditional numerical methods like FEM and FDM in handling highly complex or large-scale scenarios highlight the compelling need for quantum enhancements. This interplay between the inherent challenges of physical phenomena and the evolving capabilities of

computational paradigms dictates a future where hybrid systems are not merely a transitional phase but a fundamental architectural model.

The integration strategies, ranging from loose networked connections to tight co-location, are continuously being refined, with advanced middleware playing a crucial role in orchestrating workloads and managing heterogeneous resources. The significant role of GPU acceleration in enhancing quantum circuit simulations and optimizing hybrid workflows further emphasizes the interdependency of these technologies.

While formidable challenges persist, including hardware limitations, the imperative for robust quantum error correction, and the complexities of software-hardware co-design and data handling, the opportunities are equally profound. The ongoing advancements in logical qubits and error mitigation techniques are steadily pushing towards fault-tolerant quantum computing, which will unlock even broader applications. The convergence with exascale computing, where quantum processors act as powerful accelerators within massive classical infrastructures, promises a future of unprecedented computational power for scientific discovery and engineering innovation. Realizing this potential demands sustained interdisciplinary collaboration, a focus on workforce development, and strategic investments to bridge the gap between theoretical promise and practical utility. The future of advanced computation for physical problems is undeniably hybrid.

References

- [1] Arora, S., & Barak, B. (2009). *Computational complexity: A modern approach*. Cambridge University Press.
- [2] Moore, G. E. (1965). Cramming more components onto integrated circuits. *Electronics*, 38(8), 114–117.
- [3] Shalf, J. (2020). The future of computing beyond Moore’s Law. *Philosophical Transactions of the Royal Society A*, 378(2166). <https://doi.org/10.1098/rsta.2019.0061>
- [4] Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman.
- [5] Nielsen, M. A., & Chuang, I. L. (2010). *Quantum computation and quantum information: 10th anniversary edition*. Cambridge University Press.
- [6] Deutsch, D., & Jozsa, R. (1992). Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907), 553–558. <https://doi.org/10.1098/rspa.1992.0167>
- [7] Ladd, T. D., Jelezko, F., Laflamme, R., Nakamura, Y., Monroe, C., & O’Brien, J. L. (2010). Quantum computers. *Nature*, 464(7285), 45–53. <https://doi.org/10.1038/nature08812>
- [8] Feynman, R. P. (1982). Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7), 467–488. <https://doi.org/10.1007/BF02650179>
- [9] Preskill, J. (2018). Quantum computing in the NISQ era and beyond. *Quantum*, 2, 79. <https://doi.org/10.22331/q-2018-08-06-79>
- [10] Bharti, K., Cervera-Lierta, A., Kyriienko, O., et al. (2022). Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1), 015004. <https://doi.org/10.1103/RevModPhys.94.015004>
- [11] Fowler, A. G., Mariantoni, M., Martinis, J. M., & Cleland, A. N. (2012). Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3), 032324. <https://doi.org/10.1103/PhysRevA.86.032324>
- [12] Moll, N., Barkoutsos, P., Bishop, L. S., et al. (2018). Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology*, 3(3), 030503. <https://doi.org/10.1088/2058-9565/aab822>
- [13] Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P. J., Aspuru-Guzik, A., & O’Brien, J. L. (2014). A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5, 4213. <https://doi.org/10.1038/ncomms5213>
- [14] Endo, S., Cai, Z., Benjamin, S. C., & Yuan, X. (2021). Hybrid quantum-classical algorithms and quantum error mitigation. *Journal of the Physical Society of Japan*, 90(3), 032001. <https://doi.org/10.7566/JPSJ.90.032001>
- [15] Bauer, C. W., de Jong, W. A., Nachman, B., & Provasoli, D. (2020). Quantum algorithms for high energy physics. *arXiv preprint arXiv:2001.03488*. <https://arxiv.org/abs/2001.03488>
- [16] Lubinski, T., Johri, S., Varner, P., et al. (2021). Calibrating and benchmarking a superconducting quantum annealing system. *arXiv preprint arXiv:2102.04207*. <https://arxiv.org/abs/2102.04207>
- [17] Shor, P. W. (1994). Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science* (pp. 124–134). <https://doi.org/10.1109/SFCS.1994.365700>

- [18] Harrigan, M. P., Sung, K. J., Neeley, M., et al. (2021). *Quantum approximate optimization of non-planar graph problems on a planar superconducting processor*. *Nature Physics*, 17, 332–336. <https://doi.org/10.1038/s41567-020-01105-y>
- [19] Zurek, W. H. (2003). *Decoherence, einselection, and the quantum origins of the classical*. *Reviews of Modern Physics*, 75(3), 715–775. <https://doi.org/10.1103/RevModPhys.75.715>
- [20] LaRose, R. (2019). *Overview and comparison of gate level quantum software platforms*. *Quantum*, 3, 130. <https://doi.org/10.22331/q-2019-03-25-130>
- [21] Zienkiewicz, O. C., Taylor, R. L., & Zhu, J. Z. (2005). *The finite element method: Its basis and fundamentals* (6th ed.). Butterworth-Heinemann.
- [22] LeVeque, R. J. (2007). *Finite difference methods for ordinary and partial differential equations: Steady-state and time-dependent problems*. SIAM. <https://doi.org/10.1137/1.9780898717839>
- [23] McCaskey, A. J., Dumitrescu, E. F., Chen, M., Ly, D. T., & Li, Y. (2020). *A language and runtime for hybrid quantum-classical computing in HPC environments*. In *2020 IEEE/ACM Second International Workshop on Quantum Computing Software (QCS)* (pp. 48-57). <https://doi.org/10.1109/QCS51206.2020.00015>
- [24] Fu, X., Riesebo, L., Lao, L., et al. (2019). *A heterogeneous quantum computer architecture*. In *2019 IEEE International Conference on Rebooting Computing (ICRC)* (pp. 1-10). <https://doi.org/10.1109/ICRC.2019.00008>
- [25] McCaskey, A. J., Claudino, D., Dumitrescu, E. F., et al. (2022). *A case for hybrid quantum-classical computing for materials science*. *Philosophical Transactions of the Royal Society A*, 380(2219). <https://doi.org/10.1098/rsta.2021.0065>