

# Quantum computing in civil engineering: Limitations

Joern Ploennigs<sup>1</sup>, Markus Berger<sup>1</sup>, Martin Mevissen<sup>2</sup>, and Kay Smarsly<sup>3</sup>

<sup>1</sup> University of Rostock, Rostock, Germany, {Joern.Ploennigs, Markus.Berger}@uni-rostock.de

<sup>2</sup> IBM Research Europe, Dublin, Ireland, martmevi@ie.ibm.com

<sup>3</sup> Hamburg University of Technology, Hamburg, Germany, Kay.Smarsly@tuhh.de

## Abstract

Quantum computing is a new computational paradigm with the potential to solve certain computationally challenging problems much faster than traditional approaches. Civil engineering encompasses many computationally challenging problems, which leads to the question of how well quantum computing is suitable for solving civil engineering problems and how much impact and implications to the field of civil engineering can be expected when deploying quantum computing for solving these problems. To address these questions, we will, in this paper, first introduce the fundamentals of quantum computing. Thereupon, we will analyze the problem classes to elucidate where quantum computing holds the potential to outperform traditional computers and, focusing on the limitations, where quantum computing is not considered the most suitable solution. Finally, we will review common complex computation use cases in civil engineering and evaluate the potential and the limitations of being improved by quantum computing.

**Keywords:** Quantum Computing, Civil Engineering.

## 1 Introduction

Civil engineering is a complex area with many challenging problems in design, construction, and operation, which leads to the adaptation of many digital technologies from computer aided design to construction robotics and machine learning, aiming to solve the underlying complex computational problems. The digitization of the industry is pushing the boundaries of what can be planned, analyzed, and optimized on current conventional computers, entailing a continuous need for new approaches.

Quantum computing (QC) is one of these new approaches. It may allow us to push some of the boundaries, as QC promises to compute numerous complex problems significantly faster than conventional computers. QC is explored in different industries [8, 25], but has not yet gotten much attention in civil engineering. The absence in civil engineering raises questions about the suitability of QC for addressing the computational challenges in civil engineering and the extent of its anticipated impact on the field. In this paper, we will review promising QC methods that may be applied to current problems in civil engineering.

Therefore, we will first introduce the fundamental principles of quantum computing. We will delve into an analysis of scenarios where QC may demonstrate an advantage over traditional computers and identify areas where its application may be far in the future. With this context in mind, we will assess prevalent complex computational problems within civil engineering, determining problem classes that qualify for enhancement through QC.

## 2 Introduction to Quantum Computing

### 2.1 Fundamentals

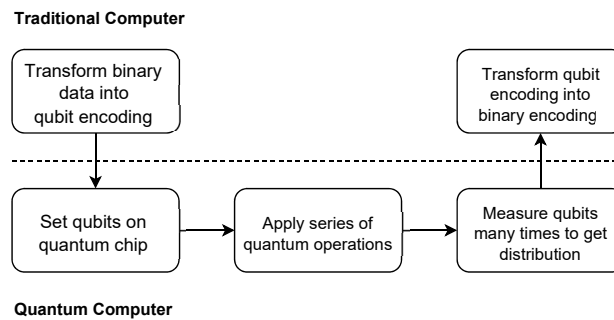
Traditional computers are based on processing and storing binary data in the form of bits that are either 0 or 1. A modern CPU is specialized on processing the binary data, usually in blocks of 64 bits, by applying logical operators (AND, OR, XOR, NOT), numerical operators (addition, subtraction, multiplication and division), or specialized operators (encoding, encryption, etc.). Well established means have been proposed to represent any kind of data in binary code, such as numbers, texts, to images, and videos. The main assumptions behind binary encoding and processing are that (i) data is encodeable in binary 0 or 1 representation and (ii) all operators are strictly deterministic.

Quantum computing, by contrast, does not follow either of these assumptions, but uses a model of computation centered on *quantum bits* or *qubits* [69]. First, a qubit can be in a complex linear combination of both basis states  $|0\rangle$

and  $|1\rangle$ , thanks to the principles of quantum *superposition*, which allows a single qubit to represent multiple states in a Hilbert space.

Secondly, quantum computers use the quantum mechanical effect of *entanglement*, where the state of one qubit becomes linked or correlated with the state of another qubit, even when separated by large distances. With those entangled qubits it is possible to create quantum gates that implement specific quantum operations as exemplified in the next section.

Third, due to the superposition principle (famously known as the Schrodinger's cat paradox), a qubit will collapse from its superposition state once measured and is identifiable only as one of the binary ground states. So, instead of being able to read the complex state between  $|0\rangle$  and  $|1\rangle$  directly, it is only possible to randomly sample binary states as either 0 or 1. Therefore, the qubit setup, computation and measurement operation needs to be repeated many times to retrieve a probability distribution representing the quantum state.



**Fig. 1.** Process for combining traditional computers with quantum computers.

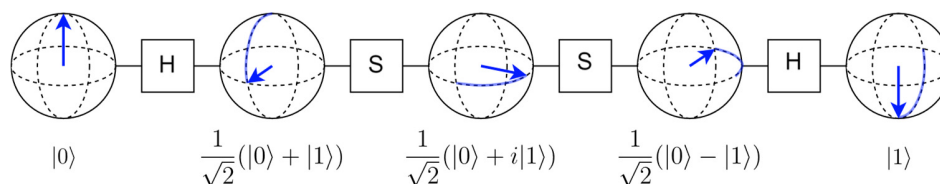
Fig. 1 summarizes the process of loading data from a traditional computer into a quantum computer, applying quantum algorithms, and subsequently iterating through result retrieval. Several architectures are currently being explored to realize universal gate-based quantum computers at scale. These include superconducting quantum computing, trapped ion systems, linear optical quantum computing and others. For example, IBM has published a development roadmap for its superconducting quantum computing systems for the next 10 years [30].

## 2.2 Introduction to quantum circuits

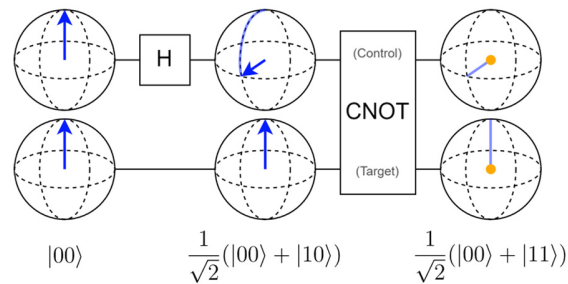
In this section we want to discuss the universal quantum gates that form quantum circuits to illustrate the complexity of quantum computation. To visualize the states of one qubit, we use the Bloch sphere model [80]. On it the states of  $|0\rangle$  or  $|1\rangle$  are positioned at opposite poles. Points off the poles represents a superposition of qubit states.

Quantum gates alter the states of the qubits on the surface of this sphere. Notable among these are one-qubit gates like the Hadamard gate (H), which initiates superposition of the initial states, and the Phase gate (S), which rotates the qubit state on the sphere through complex space [60]. With these gates we can build a quantum circuit to negate a single qubit from  $|0\rangle$  to  $|1\rangle$  as shown in Fig. 2.

Another essential gate is the CNOT (Controlled-NOT-Gate) gate that operates on two qubits. It allows us to manipulate the state of one qubit based on the state of a second qubit. Specifically, the value of the second qubit (target) is either retained ( $|00\rangle \rightarrow |00\rangle$ ;  $|01\rangle \rightarrow |01\rangle$ ) or negated ( $|10\rangle \rightarrow |11\rangle$ ;  $|11\rangle \rightarrow |10\rangle$ ) depending on whether the first qubit (control) is  $|0\rangle$  or  $|1\rangle$ , respectively. If the control is in a state of superposition, this results in the qubits getting entangled, as the state of the target is entirely dependent on how the superposition of the control resolves. Fig. 3 shows an example.



**Fig. 2.** A quantum circuit with a series of Hadamard gates (H) and Phase gate (S) to move a qubit from a pure  $|0\rangle$  to a pure  $|1\rangle$  state, shown as vectors on a Bloch sphere.



**Fig. 3.** A quantum circuit with two qubits. The Hadamard gate (H) places the control qubit of the CNOT gate into a superposition state, resulting in the qubits becoming fully entangled.

The CNOT gate and all one-qubit gates like the Hadamard and Phase gate form the set of universal gates that allow us to implement all other quantum gates and quantum algorithms. The resulting state of the target qubits at the end of the operation is measured multiple times, as discussed before.

### 3 Challenges

There is a lot of ongoing fundamental research in quantum computing, and many challenges exist that limit its application at scale in practice. In this section, we aim to address some of the most prominent challenges, while dispelling common misconceptions about the current readiness of QC.

**Error handling:** Compared to classical computers, there are many sources of interference in quantum computing hardware, also called noise. This noise constrains the number of operational qubits. Current quantum computers are, therefore, *noisy quantum computers* [68]. As a result, one of the most prominent challenges in scaling up quantum computers is handling these errors affecting the actual quantum states and qubits. Broadly, we distinguish two approaches: Error mitigation and Error correction [12].

*Error mitigation* uses the outputs of quantum circuits to reduce the effect of noise in estimating expectation values and is an approach to improve the performance of current noisy quantum computer in the short term. Two state-of-the-art techniques for error mitigation are Probabilistic Error Cancellation [78] and Zero-Noise Extrapolation [39]. Those approaches estimate the noise and introduce additional computational overhead with limited scalability.

The long-term research targets Fault-Tolerant universal Quantum Computers (FTQC) by using error correction. *Error correction* is based on built-in redundancies, that is, on encoding each logical qubit through multiple redundant physical qubits [67] and running error correction code for detecting and correcting errors. The redundancy, while critical for error correction, poses a significant challenge to the potential scalability of FTQC, as solving practical problems may require millions of qubits [14]. Thus, it is expected to take a number of years before first FTQC can be realised. Therefore, both approaches feature a trade-off between the additional overheads introduced by running the error mitigation or correction method on the one hand and the reduction in noise achieved on the other.

**Quantum Compilation:** Over the past few decades, we have become accustomed to the convenience of enhancing our computational performance by either upgrading hardware components or adding more machines due to standardized hardware, like the IBM PC, and operating systems, such as Linux, Windows, or iOS. This familiarity has led to an expectation that transitioning to quantum computing should be equally straightforward, allowing us to run our existing applications on quantum computers for vastly improved performance.

As shown in the introduction, quantum computers operate on fundamentally different principles than traditional computers, in terms of problem representation and solution methods. Consequently, we cannot simply cross-compile existing code to be run on a quantum computer in the same manner as we would do with a new CPU. Instead, each problem must be specifically adapted and implemented for quantum computation. Special programming languages for QC, such as IBM Qiskit, have been developed over the past few years [40]. Nevertheless, the development of efficient quantum compilers remains a significant challenge, with many approaches under investigation on how to optimization exact or approximate quantum compilation [57, 58]. As a result, quantum computing programming languages, while resembling modern high-level programming languages, often operate at a very low level, close to the quantum gates themselves, and require a profound understanding of the underlying quantum computing models. Thankfully, experts develop problem specific frameworks, e.g. for quantum chemistry, machine learning or optimization problems that lower the entry barrier to utilize quantum computers for those use cases.

The development of GPU computing offers some parallels that can provide valuable insights. While modern PCs are equipped with GPUs boasting hundreds, if not thousands, of cores—far more than traditional CPUs—GPUs have not replaced CPUs because many computational problems in computing are not *embarrassingly parallelizable* and, thus, well-suited for GPUs. Instead, most problems are highly sequential, allowing the faster CPUs to solve them much more efficiently. Parallelizable problems also need to be specially implemented for the GPU in dedicated programming languages or frameworks. For instance, NVIDIA introduced CUDA in 2007 [50] with support for many programming languages. Yet, its applications in civil engineering software remain limited [73]. Instead, it took several years until the area of neural networks took advantage of it to scale deep learning, which now also has a disruptive impact on civil engineering.

**Quantum advantage:** Most existing software runs efficiently on traditional computers, both from a computational and from an energy perspective [44]. For certain specific problem classes quantum computers have the potential to outperform classical computers in the near term. This is called *quantum advantage*, when we are able to demonstrate that the quantum computer can outperform classical computers and find problem instances where this speedup is useful. Early quantum algorithms, such as Deutsch–Jozsa [22], Shor [72], and Grover [33], were ground setting for this research area, but, have limited direct applicability on their own. The current focus is on finding problems where noisy quantum computers may provide computational advantage over classical computers considering that most algorithms require error correction, for example the widely re-used Shor algorithm for integer factorization. Indeed, there have been recent results demonstrating the utility of quantum computing for simulating the time evolution of 2D transverse-field Ising models [49]. There is a lot of ongoing research at the moment, in order to find further challenging problems, where noisy quantum computers—along with techniques like error mitigation—are demonstrated to be a useful tool.

## 4 Use Cases

### 4.1 Use Cases for Quantum Computing

In this section, we will explore use cases that are expected to exhibit a quantum advantage in the near future and share similarities with problems encountered in civil engineering. The use cases encompass: (i) *simulating* natural physical processes, (ii) *mathematical and machine learning* problems that process data with complex structure, and (iii) solving *optimization* problems as well as search.

**Simulating** nature represents one significant application area for quantum computers, particularly for systems operating (at or close to) the quantum level, such as simulating Hamiltonian dynamics or preparation of ground states. There is an increasing body of work that studies the properties of many body quantum systems with the help of quantum computers, e.g. [47]. Current exploration in the field includes applications in chemistry, such as the study of molecular structures [62] or chemical reactions [13]. Quantum computers facilitate the investigation of complex materials, including superconductors and novel compounds, allowing for more efficient comprehension of the properties and potential applications compared to classical methods. Leveraging the precision in simulating quantum mechanics, quantum computers are well-suited for providing more accurate predictions of molecular behavior, which is promising for future developments in drug discovery, materials design, and catalyst development [17]. A particularly encouraging recent result has been the demonstration of the utility of quantum computing for simulating the time evolution of 2D transverse-field Ising models [49].

Certain **Mathematical and Machine Learning** problems that process data with complex structure are expected to benefit from approaches based on quantum computing. In particular, demonstrating exponential speed-up is expected for quantum machine learning methods. While theoretic development of quantum machine learning algorithms goes back several decades, there has been a growing body of work on quantum machine learning approaches explored on current noisy quantum computers.

A promising class of quantum machine learning approaches are quantum kernel methods for classification problems, one of the most fundamental problems in machine learning. In a recent result, it was shown that there are specific classification tasks where a quantum kernel method with only classical access to data provides an exponential speed-up over classical machine learning algorithms [55]. It is an ongoing direction of research to develop supervised quantum machine learning algorithms for broader sets classification tasks.

Another notable group of quantum algorithms with potential applications in machine learning includes quantum phase estimation (QPE) [24] and the underlying quantum Fourier transform (QFT) [34], which is similar to the classical discrete Fourier transform and performs a Fourier transformation of the amplitudes but scales in  $O(n^3)$  whereby the best-known classical algorithm requires  $2^{O(n^{1/3})}$ . QFT finds widespread usage in various other algorithms, including

Shor's algorithm for integer factoring [72], the Harrow-Hassidim-Lloyd (HHL) algorithm for solving linear equations [36], and quantum gradient estimation [45]. One application area of Shor's algorithm is cryptography, where the algorithm enables efficient factorization of integers to find the prime factors.

The advancements made in solving specific mathematical matrix problems using QAOA have found applications in enhancing traditional machine learning models, including linear regression [20], clustering [31, 42], reinforcement learning [23, 52], support vector machines [38], and active learning [63]. Additionally, there is ongoing exploration of quantum algorithms for neural networks [48, 70]. Furthermore, quantum computing has been investigated for its potential to improve classical machine learning algorithms, particularly in the context of handling tensor and dot products in higher dimensions, thereby reducing computation time [19, 27]. In all these cases, quantum computing addresses specific aspects of traditional algorithms.

A sub-category of the aforementioned problems includes classical data analysis and statistics problems, which can be advanced through machine learning algorithms and, also, by more classically adapted methods, such as quantum principal component analysis [56] or quantum clustering [3]. Again, quantum advantage has not been shown to be practical for any of these models, yet, and achieving it on current noisy quantum computers is challenging due to the necessity of conducting numerous measurement runs and the stochastic nature of the objective function.

**Optimization** is another class of problems where quantum computing may be demonstrated to be useful, even if no exponential speed-up over classical computing is expected. However, various problems may also derive benefits from quantum computing [1]. Several gate-based optimization algorithms have been developed, involving the simulation of the evolution of a system through a sequence of quantum operators. Two common methods include the variational quantum eigensolver (VQE) [66] and the quantum approximate optimization algorithm (QAOA). Both methods are meta-heuristics for solving combinatorial optimization problems that can leverage gate-based quantum computers and potentially outperform purely classical heuristic algorithms [29]. The approaches mentioned above find application in logistical problems [5, 79], such as optimizing traffic networks to reduce congestion and enhance efficiency [81] or formulating and solving routing problems [37]. Similarly, the methods are employed for optimizing energy distribution networks [4]. A common feature among these applications is the utilization of a graph representation [32], where the problems are modeled as a Max-Cut problem.

Finding an exact solution to the Max-Cut problem is known to be NP-hard [46]. The objective of the Max-Cut problem is to divide the set of graph vertices into two subsets in a way that the sum of the weights of the edges connecting one subset to the other is maximized. Solving Max-Cut with QAOA has seen some focus since it can be formulated as an unconstrained quadratic unconstrained binary optimization (QUBO) problem and has wide number of applications. The approach with QAOA is based on the following mapping:

$$\arg \min_x \sum_{i,j} w_{i,j} x_i x_j \xrightarrow{\text{mapped to QAOA}} H = \sum_{i,j} w_{i,j} \sigma_i \sigma_j$$

where  $w_{i,j}$  are the weights for the binary assignment  $x_i$  and  $x_j$ . To solve the problem using QAOA, we construct the cost Hamiltonian  $H$  by mapping the binary assignment variables  $x_k$  onto the eigenvalues of the Pauli Z operator  $\sigma_k$ . By employing the mapping, we obtain a representation on the right that closely resembles the original problem. It is expected, albeit unproven, that a quantum computer can solve this problem more efficiently than a classical one [71].

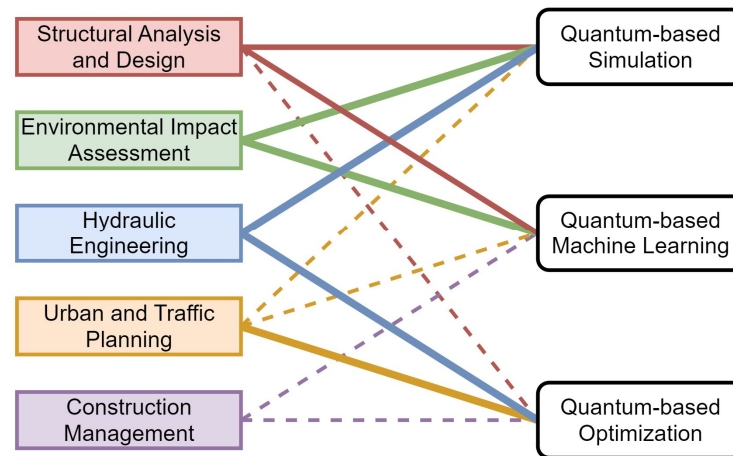
## 4.2 Use cases in civil engineering

Based on this discussion of use cases for quantum computers we can derive some insights on which use cases in civil engineering may be most appropriate to benefit from quantum computers. We derive the mapping shown in Fig. 4 with good (bold), some (dotted) and limited (none) applicability as explained below.

**Simulations** are a common approach in civil engineering to evaluate design variants or to understand real-world phenomena. Some of those approaches may benefit from quantum computing.

*Urban and traffic planning* is using simulations specifically for modeling people and traffic flows and congestion within transportation networks, encompassing interactions among vehicles, infrastructure, and traffic signals. Traffic simulations rely on computational models that account for a multitude of variables, ranging from vehicle behavior and road conditions to traffic patterns [77]. These simulation models are less suited for quantum.

*Structural analysis and design* is commonly employing either rigid body models or finite element analysis (FEA) to simulate and analyze the behavior of structures under various conditions, including stress, heat transfer, and fluid flow. FEA, in particular, demands substantial computational resources as it involves dividing a structure into numerous small



**Fig. 4.** An overview of engineering disciplines associated with the potential applicability of quantum computing. Lines highlight good (bold), some (dotted), or limited (none) applicability.

interacting elements to approximate the overall system behavior [82]. *Geotechnical engineering* problems are another application area for FEA. Here, engineers simulate soil-structure interactions, slope stability analysis, and foundation design [18]. *Seismic analysis and design* combines aspects of both structural and geotechnical engineering and focuses on designing structures capable of withstanding seismic forces, entailing complex analyses of how structures respond to earthquakes [43].

*Hydraulic engineering* often employs computational fluid dynamics (CFD) models to analyze the behavior of rivers, dams, coastal areas, and other hydraulic structures. The models, like FEA, break down problems into small interacting fluid (or gas) elements, but emphasize the solution of intricate fluid dynamics equations such as turbulence modeling, sediment transport, and interactions with structures [6].

*Environmental Impact Assessment* is looking into simulating the environmental impact of constructions often applying a combination of CFD or FEA methods [35].

The FEA and CFD problems fundamentally involve simulating the underlying physics in the form of higher-order differential equations, without relying on specific quantum mechanical effects for which quantum computers are well-suited. Instead, only quantum algorithms that enhance the efficiency of solving differential equations are explored [11, 54]. They break them down into systems of linear equations solvable with quantum algorithms, as the HHL [36].

A few applications to civil engineering problems can be found. Ajagekar and You [4] show a heat exchanger network synthesis problem, where multiple cooling units need to be balanced. Quantum approaches are utilized to simulate the behaviour and to identify the optimal solution.

Various **Mathematical and Machine Learning** problems exist in civil engineering, primarily in operation scenarios, but also in design and construction.

*Structural analysis and design* is using methods for *structural health monitoring* to record and analyze sensor data for assessing structural conditions of infrastructure, including the detection of defects or damage, aiming to support predictive maintenance and life-cycle management [28, 74]. *Predictive maintenance* is a related topic that focuses on analyzing system performance to predict maintenance actions [15], applicable not only to structures but to various types of assets.

*Environmental Impact Assessment* is another field using prediction models to analyze performance measures in general for a system like their energy consumption [2]. *Performance-based design* can conceptually use the models to evaluate different design options under various conditions or environmental impacts. If the prediction models are trained from simulation models, we speak of *surrogate models* [75].

*Hydraulic and Energy Engineering* is commonly targeting *demand prediction* of consumers in energy or water grids, or the traffic flow of people and cars in *Urban and Traffic Planning*. The objectives are similar to the simulation scenarios discussed earlier but rely on data-driven models instead of simulation models [53].

The specific machine learning models employed depend on the use case and data type. Typically, the data is recorded through sensors and stored in the form of time series. Regression models, such as linear regression or support vector machines, are often sufficient for prediction [65]. Quantum computing variants exist for these models, as discussed earlier [20, 38], but their necessity is debatable, as time series problems can be decomposed and trained individually at

larger scale [64]. For problems with a graph structure, such as energy, water or traffic networks, Graph Neural Networks may be more appropriate [7], although more performant quantum variants are currently lacking. In cases involving image or video data sets, deep learning models are commonly used, as the models tend to benefit more from GPU processing than quantum computation do [16, 21].

**Optimization** in civil engineering involves finding the best arrangement, combination, or configuration of elements within a discrete set of choices, considering various constraints and objectives.

*Construction Management* deals with efficiently assigning resources (e.g. labor, equipment, materials) in allocation and scheduling problems and schedule construction activities [41]. These problems can often be modelled as a QUBO problem [51].

*Urban and traffic planning* addresses the design of optimal traffic networks [76], which is an extension of the logistic routing problem discussed earlier [5, 79]. However, there has not been a discussion yet on the evaluation of the impact of the algorithms on network design. Conceptually related is the *site location selection* problem, which refers to determining the most suitable locations for infrastructure facilities, such as schools, hospitals, waste management sites, or transportation hubs. The determinations take into account factors, such as accessibility, environmental impact, population distribution, and cost [61]. The problems can often be modeled as coverage problems [26], and thus we may be able to map the problems onto a QUBO problem, which, in turn, might be solved by QAOA. However, in practice, the number of constraints and weights is often higher, as demonstrated by Farahani et al. [26].

*Hydraulic engineering* and distribution networks planning problems are prevalent in civil engineering, spanning various forms from power to fresh and wastewater networks. Early work explores the use of quantum computers for optimizing energy distribution networks [4]. Similar approaches may apply to other network types by mapping the problems to graph Max-Cut problems.

*Environmental Impact Assessment* is usually more looking in using optimization for improving energy efficiency. Here the optimization approaches are either control problems or scheduling problems.

*Structural analysis and design* is optimizing the arrangement and sizing of structural components (beams, columns) that ensure structural integrity while minimizing material usage [9, 59]. Optimization often requires solving complex mathematical problems, such as nonlinear programming or genetic algorithms. Considering the problem as a rigid body problem with a graph representation may allow to map it to QAOA. For example, Wang et al. [79] discuss an application for a topology optimization problem called the Messerschmitt-Bolkow-Blohm (MBB) problem [10] to find the stiffest design of a desired volume fraction. Traditionally, such design problems were solved by engineers through experience or trial and error.

## 5 Conclusion

In this paper, we have discussed the current state of quantum computing and potential applications in civil engineering. We first provided an overview of the fundamental concepts of quantum computing, highlighting its distinctions from traditional binary computing. Subsequently, we have addressed the current challenges associated with scaling quantum computing, with a specific focus on algorithms and problem classes that can be studied with current, noisy quantum computers. We then have reviewed the areas where quantum computing has the potential to outperform traditional computing, namely in simulating quantum mechanics, quantum machine learning approaches, and classes of optimization problems. For each area, we have also introduced the relevant use cases in civil engineering and have discussed the potential of applying QC within them. The preliminary conclusions are summarized as follows:

- **Simulation:** Among the most computationally demanding tasks are FEA and CFD problems, where quantum computing holds the potential to enhance performance in solving differential equations. Here, the discretization into linear equation systems is an important step.
- **Machine Learning:** Quantum computing shows the largest promise in the speed-up that may be achieved through quantum kernel methods for a variety of classification tasks. In addition, advancements in machine learning involve algorithmic steps that may have the potential to perform better on a quantum computer. To benefit from these future possibilities, it is recommended to keep investigating the applicability of these classification models in civil engineering.
- **Optimization:** Quantum computing has already been explored for a number of optimization problems arising across different application domains. There is a lot of ongoing research in the speed up that may be achieved for particular algorithms. In civil engineering this means to identify relevant problem-mappings to solutions like Max-Cut or QUBO.

Quantum computing has the potential to catalyze a revolutionary paradigm shift similar to the transformative impact observed with GPU computing, currently reshaping the landscape of science through the empowerment of deep learning. The innovations in quantum computing will not arise primarily from accelerating existing code; instead, innovations are expected to derive from the ability to address an entirely new set of challenges using specialized software applicable across a broad spectrum of domains.

## References

1. A. Abbas, A. Ambainis, B. Augustino, A. Bartschi, H. Buhrman, and et al. Quantum Optimization: Potential, Challenges, and the Path Forward. arXiv e-prints, art. arXiv:2312.02279, Dec. 2023.
2. A. Ahmed, N. E. Korres, J. Ploennigs, H. Elhadi, and K. Menzel. Mining building performance data for energy-efficient operation. *Advanced Engineering Informatics*, 25(2):341–354, 2011.
3. E. Aïmeur, G. Brassard, and S. Gambs. Quantum clustering algorithms. In *ICML*, pages 1–8, 2007.
4. A. Ajagekar and F. You. Quantum computing for energy systems optimization: Challenges and opportunities. *Energy*, 179: 76–89, 2019.
5. A. Ajagekar, T. Humble, and F. You. Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems. *Computers & Chemical Engineering*, 132:106630, 2020.
6. D. Anderson, J. C. Tannehill, R. H. Pletcher, R. Munipalli, and V. Shankar. *Computational fluid mechanics and heat transfer*. 2020.
7. A. Ba, F. O’Donncha, J. Ploennigs, and M. Azmat. Efficient extraction of insights at the edges of distributed systems. In *IEEE BigData*, 2023.
8. A. Bayerstadler, G. Becquin, J. Binder, T. Botter, H. Ehm, T. Ehmer, M. Erdmann, N. Gaus, P. Harbach, M. Hess, et al. Industry quantum computing applications. *EPJ Quantum Technology*, 8(1):25, 2021.
9. A. D. Belegundu and T. R. Chandrupatla. *Optimization concepts and applications in engineering*. 2019.
10. M. P. Bendsoe and O. Sigmund. *Topology optimization: theory, methods, and applications*. 2003.
11. D. W. Berry. High-order quantum algorithm for solving linear differential equations. *Journal of Physics A: Mathematical and Theoretical*, 47(10):105301, 2014.
12. Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O’Brien. Quantum error mitigation. *Reviews of Modern Physics*, 95(4):045005, 2023.
13. Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferova, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. Sawaya, et al. Quantum chemistry in the age of quantum computing. *Chemical reviews*, 119(19), 2019.
14. A. Chatterjee, P. Stevenson, S. De Franceschi, A. Morello, N. P. de Leon, and F. Kuemmeth. Semiconductor qubits in practice. *Nature Reviews Physics*, 3(3):157–177, 2021.
15. J. C. Cheng, W. Chen, K. Chen, and Q. Wang. Data-driven predictive maintenance planning framework for MEP components based on BIM and IoT using machine learning algorithms. *Automation in Construction*, 112, 2020.
16. E. F. Combarro, S. Gonzalez-Castillo, and A. Di Meglio. *A Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands-on Approach to Modern Quantum Algorithms*. 2023.
17. A. J. Daley, I. Bloch, C. Kokail, S. Flannigan, N. Pearson, M. Troyer, and P. Zoller. Practical quantum advantage in quantum simulation. *Nature*, 607(7920):667–676, 2022.
18. B. M. Das. *Principles of geotechnical engineering*. 2021.
19. V. R. Dasari, M. S. Im, and L. Beshaj. Solving machine learning optimization problems using quantum computers. In *Disruptive Technologies in Information Sciences IV*, volume 11419, pages 60–69, 2020.
20. P. Date and T. Potok. Adiabatic quantum linear regression. *Scientific reports*, 11(1):21905, 2021.
21. P. Date, C. Schuman, R. Patton, and T. Potok. A classical-quantum hybrid approach for unsupervised probabilistic machine learning. In *Future of Inf. and Com. Conf. (FICC)*, pages 98–117, 2020.
22. D. Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *Proc. of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.
23. D. Dong, C. Chen, H. Li, and T.-J. Tarn. Quantum reinforcement learning. *IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(5):1207–1220, 2008.
24. U. Dorner, R. Demkowicz-Dobrzanski, B. J. Smith, J. S. Lundeen, W. Wasilewski, K. Banaszek, and I. A. Walmsley. Optimal quantum phase estimation. *Physical review letters*, 102(4):040403, 2009.
25. D. J. Egger, C. Gambella, J. Marecek, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain. Quantum computing for finance: State-of-the-art and future prospects. *IEEE Trans. on Quantum Engineering*, 1:1–24, 2020.
26. R. Z. Farahani, N. Asgari, N. Heidari, M. Hosseininia, and M. Goh. Covering problems in facility location: A review. *Computers & Industrial Engineering*, 62(1):368–407, 2012.
27. D. V. Fastovets, Y. I. Bogdanov, B. I. Bantysh, and V. Lukichev. Machine learning methods in quantum computing theory. *Int. Conf. on Micro- And Nano-Electronics 2018*, 2019. doi: 10.1117/12.2522427.



28. M. Flah, I. Nunez, W. Ben Chaabene, and M. L. Nehdi. Machine learning algorithms in civil structural health monitoring: A systematic review. *Archives of computational methods in engineering*, 28, 2021.
29. F. G. Fuchs, K. O. Lye, H. Møll Nilsen, A. J. Stasik, and G. Sartor. Constraint preserving mixers for the quantum approximate optimization algorithm. *Algorithms*, 15(6):202, 2022.
30. J. Gambetta. IBM’s roadmap for scaling quantum technology. *IBM Research Blog* (September 2020), 2020.
31. L.-Z. Gao, C.-Y. Lu, G.-D. Guo, X. Zhang, and S. Lin. Quantum k-nearest neighbors classification algorithm based on mahalanobis distance. *Frontiers in Physics*, 10:1047466, 2022.
32. T. D. Goodrich, B. D. Sullivan, and T. S. Humble. Optimizing adiabatic quantum program compilation using a graph-theoretic framework. *Quantum Information Processing*, 17:1–26, 2018.
33. L. K. Grover. A fast quantum mechanical algorithm for database search. In *An. ACM symposium on Theory of computing*, pages 212–219, 1996.
34. L. Hales and S. Hallgren. An improved quantum fourier transform algorithm and applications. In *An. Symposium on Foundations of Computer Science*, pages 515–525, 2000.
35. V. Harish and A. Kumar. A review on modeling and simulation of building energy systems. *Renewable and Sustainable Energy Reviews*, 56:1272–1292, 2016. ISSN 1364-0321.
36. A. W. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.
37. S. Harwood, C. Gambella, D. Trenev, A. Simonetto, D. Bernal, and D. Greenberg. Formulating and solving routing problems on quantum computers. *IEEE Trans. on Quantum Engineering*, 2:1–17, 2021.
38. V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, 2019.
39. A. He, B. Nachman, W. A. de Jong, and C. W. Bauer. Zero-noise extrapolation for quantum-gate error mitigation with identity insertions. *Physical Review A*, 102(1):012426, 2020.
40. B. Heim, M. Soeken, S. Marshall, C. Granade, M. Roetteler, A. Geller, M. Troyer, and K. Svore. Quantum programming languages. *Nature Reviews Physics*, 2(12):709–722, 2020.
41. J. Hinze. *Construction planning and scheduling*, volume 228. 2004.
42. D. Horn and A. Gottlieb. Algorithm for data clustering in pattern recognition problems based on quantum mechanics. *Physical review letters*, 88(1):018702, 2001.
43. J. Humar. *Dynamics of structures*. 2012.
44. D. Jaschke and S. Montangero. Is quantum computing green? An estimate for an energy-efficiency quantum advantage. *Quantum Science and Technology*, 8(2):025001, 2023.
45. S. P. Jordan. Fast quantum algorithm for numerical gradient estimation. *Physical review letters*, 95(5), 2005.
46. R. M. Karp. *Reducibility among combinatorial problems*. 2010.
47. N. Keenan, N. F. Robertson, T. Murphy, S. Zhuk, and J. Goold. Evidence of kardar-parisi-zhang scaling on a digital quantum simulator. *npj Quantum Information*, 9(1):72, 2023.
48. I. Kerendis, J. Landman, and A. Prakash. Quantum algorithms for deep convolutional neural networks. *arXiv preprint arXiv:1911.01117*, 2019.
49. Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. Van Den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, et al.: Evidence for the utility of quantum computing before fault tolerance. *Nature*, 618(7965):500–505, 2023.
50. D. Kirk et al. NVIDIA CUDA software and GPU parallel computing architecture. In *ISMM*, volume 7, 2007.
51. G. A. Kochenberger and F. Glover. A unified framework for modeling and solving combinatorial optimization problems: A tutorial. *Multiscale optimization methods and applications*, pages 101–124, 2006.
52. L. Lamata. Basic protocols in quantum reinforcement learning with superconducting circuits. *Scientific reports*, 7(1):1609, 2017.
53. Y. Lassoued, J. Monteil, Y. Gu, G. Russo, R. Shorten, and M. Mevissen. A hidden markov model for route and destination prediction. In *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017.
54. J.-P. Liu, H. Ø. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, and A. M. Childs. Efficient quantum algorithm for dissipative nonlinear differential equations. *Proc. of the National Academy of Sciences*, 118(35), 2021.
55. Y. Liu, S. Arunachalam, and K. Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, 17(9):1013–1017, 2021.
56. S. Lloyd, M. Mohseni, and P. Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, 2014.
57. L. Madden and A. Simonetto. Best approximate quantum compiling problems. *ACM Trans. on Quantum Computing*, 3(2):1–29, 2022.
58. N. Mariella and S. Zhuk. A doubly stochastic matrices-based approach to optimal qubit routing. *Quantum Information Processing*, 22(7):264, 2023.
59. L. Mei and Q. Wang. Structural optimization in civil engineering: a literature review. *Buildings*, 11(2):66, 2021.
60. M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*. 2010.
61. F. O’Donncha, J. Sheehan, F. Lorenzi, and J. Ploennigs. Large language models and knowledge graphs to optimize commercial real estate portfolio. In *INFORMS An. Meeting*, 2023.
62. C. Outeiral, M. Strahm, J. Shi, G. M. Morris, S. C. Benjamin, and C. M. Deane. The prospects of quantum computing in computational molecular biology. *WIREs Computational Molecular Science*, 11(1):e1481, 2021.

63. G. D. Paparo, V. Dunjko, A. Makmal, M. A. Martin-Delgado, and H. J. Briegel. Quantum speedup for active learning agents. *Physical Review X*, 4(3):031002, 2014.
64. D. Patel, S. Lin, D. Shah, S. Jayaraman, J. Ploennigs, A. Bhamidipati, and J. Kalagnanam. Ai model factory: scaling ai for industry 4.0 applications. In *AAAI*, volume 37, pages 16467–16469, 2023.
65. N. Pathak, A. Ba, J. Ploennigs, and N. Roy. Forecasting gas usage for big buildings using generalized additive models and deep learning. In *IEEE SMARTCOMP*, pages 203–210, 2018.
66. A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):4213, 2014.
67. L. Postler, S. Heußen, I. Pogorelov, M. Rispler, T. Feldker, M. Meth, C. D. Marciniak, R. Stricker, M. Ringbauer, R. Blatt, et al. Demonstration of fault-tolerant universal quantum gate operations. *Nature*, 605(7911), 2022.
68. J. Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018.
69. J. Preskill. Quantum computing 40 years later. In *Feynman Lectures on Computation*, pages 193–244. 2023.
70. M. Schuld, I. Sinayskiy, and F. Petruccione. The quest for a quantum neural network. *Quantum Information Processing*, 13: 2567–2586, 2014.
71. R. Shaydulin and Y. Alexeev. Evaluating quantum approximate optimization algorithm: A case study. In *Int. green and sustainable computing Conf. (IGSC)*, pages 1–6, 2019.
72. P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *An. Symposium on Foundations of Computer Science*, pages 124–134, 1994.
73. B. G. Simpson, M. Zhu, A. Seki, and M. Scott. Challenges in gpu-accelerated nonlinear dynamic analysis for structural systems. *Journal of Structural Engineering*, 149(3):04022253, 2023.
74. K. Smarsly, K. Dragos, and J. Wiggensbrock. Machine learning techniques for structural health monitoring. In *8th Eu. Workshop on Structural Health Monitoring*, 2016.
75. H. Sun, H. V. Burton, and H. Huang. Machine learning applications for building structural design and performance assessment: State-of-the-art review. *Journal of Building Engineering*, 33:101816, 2021.
76. P. Toth and D. Vigo. *Vehicle routing: problems, methods, and applications*. 2014.
77. M. Treiber and A. Kesting. *Traffic flow dynamics. Traffic Flow Dynamics: Data, Models and Simulation*, Springer-Verlag Berlin Heidelberg, pages 983–1000, 2013.
78. E. Van Den Berg, Z. K. Mineev, A. Kandala, and K. Temme. Probabilistic error cancellation with sparse pauli–lindblad models on noisy quantum processors. *Nature Physics*, pages 1–6, 2023.
79. Y. Wang, J. E. Kim, and K. Suresh. Opportunities and challenges of quantum computing for engineering optimization. *Journal of Computing and Information Science in Engineering*, 23(6), 2023.
80. H. Wiseman and G. Milburn. Interpretation of quantum jump and diffusion processes illustrated on the bloch sphere. *Physical Review A*, 47(3):1652, 1993.
81. S. Yarkoni, F. Neukart, E. M. G. Tagle, N. Magiera, B. Mehta, K. Hire, S. Narkhede, and M. Hofmann. Quantum shuttle: traffic navigation with quantum computing. In *ACM SIGSOFT Int. Workshop on Architectures and Paradigms for Engineering Quantum Software*, pages 22–30, 2020.
82. O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The finite element method: its basis and fundamentals*. 2005.