National Technical University of Athens
School of Electrical and Computer Engineering
Data Science and Machine Learning

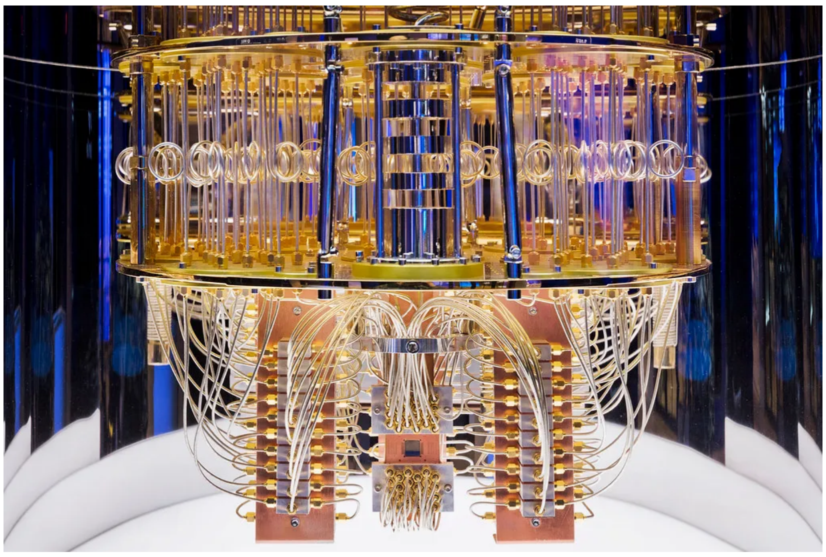
# Solving parametrized Linear Systems of equations with the Variational Quantum Linear Solver

*A first attempt to pair a Variational Quantum Algorithm with Machine Learning*

Diploma Thesis

of

**CONSTANTINOS ATZARAKIS**



**Supervisor:** V. Papadopoulos
Associate Professor

March 21, 2023

National Technical University of Athens
School of Electrical and Computer Engineering
Data Science and Machine Learning

# Solving parametrized Linear Systems of equations with the Variational Quantum Linear Solver

*A first attempt to pair a Variational Quantum Algorithm with Machine Learning*

Diploma Thesis

of

**CONSTANTINOS ATZARAKIS**

**Supervisor:** V. Papadopoulos
Associate Professor

Approved by the examination committee on 14th March 2023.

| | | |
|---|---|---|
| *(Signature)* | *(Signature)* | *(Signature)* |
| ..................... | ..................... | ..................... |
| V. Papadopoulos | G. Goumas | M. Fragiadakis |
| Associate Professor | Associate Professor | Associate Professor |

March 21, 2023

National Technical University of Athens
School of Electrical and Computer Engineering
Data Science and Machine Learning

**DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS**

*(Signature)*

. . . . . . . . . . . . . . . . . . . . . . . . . .
Constantinos Atzarakis
26th September 2022

# Abstract

The potential of quantum computing for scientific and industrial breakthroughs is immense, however, we are still in the Noisy Intermediate-Scale quantum (NISQ) era, where the currently available quantum devices contain small numbers of qubits, are very sensitive to environmental conditions and prone to quantum decoherence. Even so, existing NISQ computers have already been shown to outperform conventional computers on specific problems and the key question is how to make use of today's NISQ devices to achieve quantum advantage in the field of computational science and engineering (CSE). In this direction, this work proposes a hybrid computing formulation by combining quantum computing with machine learning for accelerating the solution of parameterized linear systems in NISQ devices. In particular, it focuses on the Variational Quantum Linear Solver (VQLS), which is hybrid quantum- classical algorithm to solve linear systems that employs a short-depth quantum circuit to efficiently evaluate a cost function related to the system solution. The circuit consists of a quantum gate sequence (unitary operators) that involves a set of tunable parameters. Then, well-established classical optimizers are being utilized to tune the parameters of the sequence so as to minimize the cost function, which is equivalent to finding the system solution at an acceptable level of accuracy. It is demonstrated in this work that we can successfully employ machine learning tools such as feed-forward neural networks and nearest-neighbor interpolation techniques to accelerate the convergence of the VQLS algorithm towards the optimal values for the circuit parameters, when applied to parameterized linear systems that need to be solved for multiple parameter instances. This of a great importance to the field of CSE as it paves the way to accelerating the solution to almost all multi-query problems (uncertainty propagation, parameter inference, optimization, sensitivity analysis etc.) as these essentially reduce to the solution of a parameterized linear system.

## Keywords

Quantum Computing, Variational Quantum Linear Solver, Quantum Linear System Problem, machine learning, parametric

*A classical computation is like a solo voice,*
*one line of pure tones succeeding each other.*
*A quantum computation is like a symphony,*
*many lines of tones interfering with one another.*
*— Seth Lloyd*

# Acknowledgements

First and foremost, I would like to express my sincere gratitude towards my supervisor Ass. Prof. Vissarion Papadopoulos for his guidance with respect to this thesis. I would also like to thank the post-doctoral researcher Yannis Kalogeris for his effort, advices and exceptional ideas which played a major role in deciding the direction of this study. Finally, I am thankful to all research associates and friends who directly or indirectly supported this modest work.

Athens, February 2023

Constantinos Atzarakis

# Table of Contents

**Abstract** **1**

**Acknowledgements** **5**

**1 Introduction** **13**

1.1 Hybrid Quantum-Classical schemes . . . . . 14

1.1.1 Linear System solution using Quantum Computers . . . . . 15

1.1.2 Solving ODEs with Quantum Computers . . . . . 16

**2 Quantum Computing - Theoretical Prerequisites** **19**

2.1 Dirac notation and Quantum Systems . . . . . 19

2.1.1 Vectors . . . . . 19

2.1.2 Vector Operations . . . . . 20

2.1.3 Density Matrices . . . . . 21

2.2 Quantum Systems . . . . . 22

2.2.1 Single-qubit System . . . . . 22

2.2.2 Multi-qubit systems . . . . . 22

2.2.3 Quantum Gates . . . . . 24

2.3 Useful Quantum Circuits for this work . . . . . 25

2.3.1 Inner Product as circuit . . . . . 25

2.3.2 The Hadamard Test . . . . . 26

2.3.3 Quantum Amplitude Estimation . . . . . 26

2.4 The Quantum Linear System Problem . . . . . 27

**3 The Variational Quantum Linear Solver** **29**

3.1 Summary . . . . . 29

3.2 Overview . . . . . 29

3.3 Cost functions . . . . . 31

3.3.1 Cost function bounds . . . . . 32

3.4 Cost evaluation . . . . . 33

3.5 Ansatz . . . . . 34

3.6 Machine Learning Enhanced VQLS for parametric systems . . . . . 35

**4 Numerical examples** **39**

4.1 Convergence and continuity . . . . . . . . . . 39
4.2 Example 1: two-qubit system . . . . . . . . . . 43
4.3 Example 2: three-qubit system . . . . . . . . . . 45
4.4 Conclusions and future work . . . . . . . . . . 48

**Bibliography** **51**

**List of Abbreviations** **53**

# List of Figures

1.1 A wafer from D-Wave's quantum annealer. . . . . . . . . . . . . . . . . . . . 13

2.1 Quantum Circuit realizing the maximally-entangled Bell state. Both qubits start from zero state. The Hadamard gate is applied on the first qubit followed by a controlled-NOT gate with the second qubit as target. Controlled operations introduce entanglement. . . . . . . . . . . . . . . . . . . . . . . . . 23

2.2 Single-qubit circuit with initial state $|0\rangle$. A Hadamard gate is applied to the qubit, followed by a Pauli-$Y$ gate. The resulting state is $|\psi\rangle = YH|0\rangle = -i/\sqrt{2}(|0\rangle - |1\rangle)$. Finally, a measurement takes place which counts whether the result state is $|0\rangle$. Due to the Hadamard gate, the probability of measuring state $|0\rangle$ is $p(0) = \left|-i/\sqrt{2}\right|^2 = 0.5$ . . . . . . . . . 24

2.3 Multi-qubit circuit with a controlled operation $CU$. $U$ operator is applied in the second $n$-qubit register only when the control (first from above) qubit has a $|1\rangle$ component. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

2.4 Circuit for Inner product. The $S^\dagger$ gate is added to the circuit when the imaginary part of the inner product is needed. The circuits $U_x, U_y$ evolve $|0\rangle^n$ to the states $|x\rangle, |y\rangle$ respectively. The white-coloured control operation implies conditional on the $|0\rangle$ state, instead of $|1\rangle$. . . . . . . . . . . . . . . . 25

2.5 Circuit for Hadamard Test. The $S^\dagger$ gate is added to the circuit when the imaginary part of the expectation value is needed. . . . . . . . . . . . . . . . 26

2.6 Circuit for original Amplitude Estimation. . . . . . . . . . . . . . . . . . . . 26

3.1 Schematic diagram of VQLS algorithm from [1]. The input to VQLS is a matrix A written as a linear combination of unitaries $A_l$ and a short-depth quantum circuit $U$ which prepares the state $|b\rangle$. The output of VQLS is a quantum state $|x\rangle$ that is approximately proportional to the solution of the linear system $Ax = b$. Parameters $\alpha$ in the ansatz $V(\alpha)$ are adjusted in a hybrid quantum-classical optimization loop until the cost $C(\alpha)$ (local or global) is below a user-specified threshold. When this loop terminates, the resulting gate sequence $V(\alpha_{\text{opt}})$ prepares the state $|x\rangle = x/\|x\|_2$, from which observable quantities can be computed. Furthermore, the final value of the cost $C(\alpha_{\text{opt}})$ provides an upper bound on the deviation of observables measured on $|x\rangle$ from observables measured on the exact solution. . . . . 30

3.2 Comparison of local $C_L$ and global $C_G$ cost performance from [1]. In all cases $\kappa = 20$. For each $n \in \{10, ..., 50\}$, the cost value versus the number of cost function evaluations is plotted. As $n$ increases it becomes increasingly hard to train to global cost function. At $n = 50$, the optimization cannot significantly lower $C_G$ below a value of one. On the other hand, $C_L$ is trainable for all values of $n$ considered. . . . . . . . . . . . . . . . . . . . . . . 32

3.3 Hadamard Test used to compute coefficients $\beta_{ll'} = \langle \mathbf{0} | V^\dagger A^\dagger_{l'} A_l V | \mathbf{0} \rangle$ [1]. The $S^\dagger$ gate is added to the circuit when the imaginary part of the expectation value is needed. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34

3.4 Hadamard Test used to compute $\zeta^i_{ll'}$ coefficients as indicated in eq. (3.22) [1]. Shown is the case for $j = 1$, when controlled-$Z$ gate is applied to first qubit of working register. The $S^\dagger$ gate is added to the circuit when the imaginary part of the expectation value is needed. . . . . . . . . . . . . . . 34

3.5 Fixed-structure layered Ansatz for $V(\mathbf{a})$ [1]. Each layer introduces $R_y(a_i)$-rotations (exploration) and controlled-$Z$ operations acting on alternating pairs of neighbouring qubits (correlation). If the linear system is restricted to real numbers, $R_y$ gates are sufficient, since they don't produce complex output. For fixed layer depth the number of variational parameters scales as $O(n)$. The example shows an ansatz with $n = 6$ qubits (linear system with size $2^6 \times 2^6$) and three layers. . . . . . . . . . . . . . . . . . . . . . . . 35

3.6 Computational workflow for the machine-learning-enhanced VQLS scheme for parametric systems. Initially, in order to produce training data, one executes the process as indicated by the solid-line arrows. Once enough training data have gathered, the machine-learning model is trained with $x^i$ as input and $\alpha^i$ as output. The computational scheme thereafter changes (indicated by the dashed arrows) as the model now proposes initial VQLS parameters $\alpha^i$, which are closer to the optimal, thus reducing the optimization cost of VQLS. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37

4.1 Optimal ansatz parameters w.r.t physical parameteres for $2 \times 2$ single-qubit system using the COBYLA method. The discontinuities in fig. 4.1a are due to the periodicity of the non-convex Hamiltonian loss which allow jumps between different local minima. In fig. 4.1b which has more than required ansatz parameters, the system becomes underdetermined resulting in more discontinuities. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41

4.2 Example of VQLS Hamiltonian loss $C_L$ for a $2^2 \times 2^2$ system per iteration. Note that BFGS algorithm uses internal iterations which are not shown in order to approximate the Hessian matrix. . . . . . . . . . . . . . . . . . . . . 42

4.3 Ansatz consisting of three parametric gates with angles $\alpha_k,\ k = 0, 1, 2$ for the $2^2 \times 2^2$ parametric QLSP example of 4.2. . . . . . . . . . . . . . . . . . 43

4.4 Pairwise joint probability distributions of optimal parameters $\alpha^i_{\text{opt},k},\ k = 0, 1, 2,\ i = 1, \ldots, 1000$ for example 4.2. . . . . . . . . . . . . . . . . . . . . . 43

4.5 Quartiles for the number of BFGS iterations required to solve an instance $A(\mathbf{p}^i)\left|x(\boldsymbol{\alpha}^i)\right\rangle = |b\rangle$ where $A(\mathbf{p}^i) \in \mathbb{C}^{4\times4}$. On the left is the pure-VQLS scheme with constant initial ansatz parameters, in the middle a similiar approach but with random initialization and on the right the naive nearby approach, where the proposed ansatz parameters for each system is taken as the optimal of the previous. The speedup factor of the latter over first two schemes based on their median values is $S \approx 1.9$. . . . . . . . . . . . . . 44

4.6 Hamiltonian loss function $C_L$ w.r.t. input ansatz parameters $\alpha_0, \alpha_1$ for example 4.2. We observe an intensive non-convex behariour and periodicity resulting in different local minima which may hinder gradient-free optimization algorithms to consistently extract continuous optimal solutions $\boldsymbol{\alpha}^i_{\text{opt}}$ across a multitude of similar linear systems. . . . . . . . . . . . . . . . . . . 45

4.7 Fixed-structure layered Ansatz with 11 parametric gates with angles $\alpha_k,\ k = 0, \dots, 10$ for the $2^3 \times 2^3$ parametric QLSP example of 4.2. . . . . . . . . . 45

4.8 Pairwise joint probability distributions of optimal parameters $\alpha^i_{\text{opt},k},\ k = 0, 1, 2,\ i = 1, \dots, 1000$ for example 4.3. In this example there are no discontinuities resulting in easier training of ML models. . . . . . . . . . . . 46

4.9 Quartiles for the number of BFGS iterations required to solve QLSP instance $A(\mathbf{p}^i)\left|x(\boldsymbol{\alpha}^i)\right\rangle = |b\rangle$, where $A(\mathbf{p}^i) \in \mathbb{C}^{8\times8}$. Like before, for the first experiment (left) the pure-VQLS scheme was with constant initial ansatz parameters, and for the second (middle) the naive nearby approach. For the last experiment (right) an MLP with 2 hidden-layers was trained with optimized ansatz parameters $\boldsymbol{\alpha}^i$ for $i = 1, \dots, N_{\text{train}} = 100$. The last two options seemingly provide the same level of acceleration with speedup factor over the original algorithm being $S \approx 2.9$ based on the median values. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47

4.10 Hamiltonian loss function $C_L$ w.r.t. input ansatz parameters $\alpha_0, \alpha_1$ for example 4.3. As in the first example, the Hamiltonian is non-convex resulting in different local minima, each one with similar loss values. . . . . . . . . 47

Chapter 1

# Introduction

Recent advancements in quantum computing are paving the way for a new age in computation, promising to revolutionize several industry standards, including communications, finance, chemistry, materials, artificial intelligence, cryptography etc., over the coming decades. Upon that realization, an informal race is currently running among governments as technological giants, startups and research centers compete in order to be the first to grasp the holy grail of quantum computing, the so-called *quantum supremacy*. When finally reached, quantum supremacy will mark a historical turning point upon which quantum processors will be able to solve problems that no classical computer can in any feasible amount of time. Though some have claimed to already have achieved this milestone, there are also counter-claims that advise modesty, since that goal is still beyond our reach.

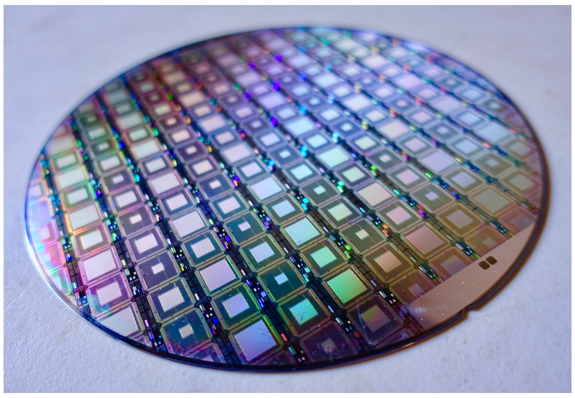


**Figure 1.1.** *A wafer from D-Wave's quantum annealer.*

Quantum technology, still in its infancy, experiences a fast growth both in theoretical level, as well as in terms of hardware and economy, as vast sums all over the world are being invested in the developing industry. Many assert that even the so-called Noisy-Intermediate Scale Quantum (NISQ) devices that we currently possess, could outperform conventional computers in the near future. Specifically, NISQ devices are near-term quantum computers, with a limited number of qubits, and too few physical qubits to implement robust error correction schemes. Existing NISQ computers have already been shown to outperform conventional computers on a limited set of problems designed specifically to fit quantum computers' capabilities.

Algorithms running on these restricted devices may require only a small number of qubits, show some degree of noise resilience, and are often cast as hybrid algorithms, where some steps are performed on a quantum device and some on a conventional computer. In particular, the number of sequential operations, or quantum gates, must remain moderate, as the longer a system is evolved, the more errors are introduced into the quantum state, and the more likely it is to decohere. Due to these restrictions, there are limits on the scope of algorithms that can be considered. The key technological question is therefore how to make best use of today's NISQ devices to achieve quantum advantage. Any such strategy must account for the limited number of qubits, toppological limitations i.e. connectivity between qubits, coherent and incoherent errors that limit quantum circuit depth.

In parallel, due to advancements in high-performance computing, which approaches fast the exascale era, the field of computational science and engineering (CSE) is also exponentially growing, being inextricably linked to the quest of solving problems of rising complexity. In such environment, quantum computing is expected to bring tremendous breakthroughs to this field. Even to this day, there are many critical problems that lay beyond our current computational capabilities and are deemed intractable even for the world's most advanced supercomputers. Some examples are non-convex optimization problems, modeling physical systems at multiple length and/or time scales, molecular modeling, robust design problems and others. With quantum computing coming to the fore, there is a silver lining in the computing world that one can finally solve such complex problems. Most field experts agree that quantum computing has the potential to bring breakthroughs to almost all aspects of CSE, from the most fundamental level of solving linear systems of equations up to machine learning modeling.

## 1.1 Hybrid Quantum-Classical schemes

As mentioned earlier, current NISQ processors suffer immensely from hardware noise, while quantum error correction — a wholesome of techniques that utilize noisy physical qubits, to create virtual, error-tolerant ones— requires many qubits, orders of magnitude more than the current availability. In principle, pure-quantum algorithms can be run on NISQ devices, however the underlying noise limits decisively the size of solvable problems. For example, HHL has been implemented with superconducting qubits [2, 3], nuclear magnetic resonance [4] etc. to a linear system of size of $2 \times 2$. Alternative approaches like an adiabatic-inspired algorithm [5] achieved on NMR a problem size of $8 \times 8$ which is — at time of writing — the current record [6].

Even according to optimistic estimations, the NISQ era won't end for at least two more decades and, while new and better quantum devices emerge continuously, gates and qubits are noisy for the time and this is something the scientific community has accepted. Along these lines, new types of hybrid algorithms have emerged which are suited for the NISQ era and combine both classical and quantum computational schemes. The main idea behind hybrid schemes is to give part or most of the higher level computations to classical CPUs, which are better-suited for such operations, and compute only critical tasks on

QPUs. Among hybrid schemes, the most notorious are the Variational Hybdrid Quantum-Classical algorithms (VHQCAs or VQAs for simplicity) [7]. VHQCAs, such as the Variational Quantum Eigensolver, employ short-depth quantum circuits — thus avoiding error accumulations— in order to efficiently evaluate a cost function, which depends on trainable parameters of a quantum gate sequence. The optimization strategy depends on well-established classical schemes which minimize the aforementioned cost. For example, Shor's algorithm for prime factorization is not well-cuited for NISQ devices. In contrast a VHQCA for factoring was introduced potentially making quantum factorizaton available sooner [8].

It is important to specify that all VHQCAs are *heuristic* algorithms, making rigorous complexity analysis rather complicated. Those familiar with machine learning, might find similarities between VHQCAs and neural networks, since both deploy similar concepts such as heuristic architectures, layers, trainable parameters, cost functions etc.

### 1.1.1 Linear System solution using Quantum Computers

CSE researchers have already started centering their attention on the development of quantum variants for classical algorithms to solve engineering problems. An important first step towards this direction can be found in the seminal work of Harrow et al. [9], where a quantum algorithm, termed HHL after its creators, was first proposed for solving linear systems of equations. While classical algorithms for solving an $N \times N$ linear system scale polynomially in $N$, HHL algorithm scales logarithmically in $N$, which suggests that quantum computers may provide exponential speedup for certain linear systems. This is a remarkable feat with major implications in CSE, since almost all problems of engineering interest are described by partial differential equations that are converted into linear systems of equations using discretization schemes (finite elements, finite differences etc.). To be more precise, the complexity of HHL, for a fixed precision $\epsilon$ in the solution, scales polynomially in $\log N$ and $\kappa$, where $\kappa$ is the condition number of the system matrix. Further improvements to HHL have been introduced lately which have reduced its complexity to linear with respect to $\kappa$ [5, 10], polylogarithmic with respect to $1/\epsilon$ [11, 12] and refined the algorithm for dense matrices [13].

In terms of real life applications, HHL-like quantum solvers have only been applied to small-scale, mostly illustrative problems such as $2 \times 2$ linear systems [14, 15]. Despite the fact that such problems could be trivially handled by classical computers, yet, these successful experiments have certainly lent much support to the growing optimism for further developments of quantum computations. The aforementioned HHL-like algorithms hold great promise for the future, when large-scale quantum computers will exist having enough qubits for quantum error correction. The timescale for such computers remains an open question, but is typically estimated to be on the order of two decades.

On the other hand, currently existing commercial quantum computers may have a few hundred noisy quantum processors, but with the maximum number of qubits rapidly increasing by the day. Thus, a crucial question is how to make use of such NISQ computers.

The most promising strategy comes from the previously mentioned VHQCAs. So far, variational schemes have been applied successfully in simulations [16], data compression [17] and metrology [18], while in [1] the authors demonstrated that they can successfully solve linear systems of medium scales. This work will focus on the last algorithm in particular, the Variational Quantum Linear Solver (VQLS) [1, 19]. The idea behind VQLS is to employ a short-depth quantum circuit to efficiently evaluate a cost function related to the system solution, but the circuit in this setting will consist of a quantum gate sequence that involves a set of tunable parameters. Then, well-established classical optimizers are being utilized to tune the parameters in the sequence so as to minimize the cost function, which is equivalent to finding the system solution at an acceptable level of accuracy.

It must be noted, that unlike pure quantum algorithms of the HHL family, VQLS shows only experimentally a polylogarithmic speed-up over classical solvers, as a theoretical complexity analysis is considered challenging and has not been realized yet.

Based on this, early theoretical investigations on the application of quantum algorithms in the context of the finite element method can already be found in the literature. In [20] authors focused on the variant of the HHL algorithm proposed by [11] and proved that the run times needed by quantum algorithms to achieve a predetermined solution accuracy could be polynomially faster than the classical ones. Similarly, in [21] the application of the HHL algorithm for the solution of finite element equations in electromagnetic problems is demonstrated and in [22] it was applied to solve the equations of elasticity. The problems studied in these works were extremely simplified, having very small dimensionality, and still quite far from practical interest. On the other hand, the VQLS algorithm shows greater promise at solving larger scale systems, as evidenced in [23] for the discretized heat equation.

### 1.1.2 Solving ODEs with Quantum Computers

Linear ordinary differential equations (ODEs) describe a plethora of phenomena in science and engineering and even nonlinear ODEs can be well approximated with linear ODEs using linearization techniques. Hence efficient solution techniques for ODEs are a focal point for researchers and scientists. In this regard, the solvers developed for linear systems of algebraic equations could also be used to solve systems of linear ODEs, as shown in [24]. However, it seems exponential speedups cannot be achieved by quantum algorithms using the linear quantum problem as a subroutine. In [25], it was shown that HHL-like algorithms used for ODEs are never faster than the best classical algorithms and an alternative approach needs to be pursued. An answer to this problem can be found in [26], where a novel gate-based quantum algorithm for solving ODEs was proposed based on the Taylor expansion of the matrix exponential, which was shown to achieve superior performance in running time. However, this idea could only be applied to problems with constant excitation terms, which is very limiting in realistic scenarios.

An alternative approach for solving ODEs by combining quantum and classical algorithms was proposed in [27]. This work manages to represent and solve high-order Runge-Kutta integration schemes as optimization problems on quantum annealers. Nevertheless, to overcome the limitations of current quantum annealers, the authors adopt the hybrid variational formulation found in the VQLS algorithm, where only a part of the problem is run on the quantum device, while a post-processing stage is performed on a classical computer to optimize the parameters in the variational algorithm. This method offers the advantage that the associated computational costs scale more favorably with increasing the integration order compared to classical computers, but it has only been demonstrated on rudimentary examples.

Chapter 2

# Quantum Computing - Theoretical Prerequisites

As Quantum Computing is at present a well-established scientific field, extensive theory shall not be covered in this thesis. As an extended introduction Quantum Information theory is out of the scope of this work, the current chapter includes only a very brief introduction to Dirac notation and Quantum Computing principles. Nevertheless, some more advanced and rather specific theoretical concepts will be discussed, since they are not part of basic quantum information theory textbooks.

For in-depth analysis and strict mathematical formulation, one of the most credible and complete sources for introductory QC concepts is [28]. Another book recommended by the author, especially for those familiar with computer science, is [29]. Besides the aforementioned sources, there is a plethora of material in the form of textbooks, lectures, courses etc. which one can follow in order to get a solid basic-level understanding of the initially peculiar concepts of Quantum Computing.

## 2.1 Dirac notation and Quantum Systems

Dirac notation is a mathematical framework derived from Quantum Mechanics and used to represent vectors, matrices, and operations on these objects. While the classical column-vector notation is common in linear algebra, it quickly becomes cumbersome in quantum computing, especially when dealing with multiple qubits. Dirac notation, in contrast, becomes handy and practical when dealing with vector subspaces and and tensor products, hence is used almost exclusively in QC. In this section we will briefly consolidate the Dirac notation how it is used to describe quantum computing systems.

### 2.1.1 Vectors

In standard notation, a vector is usually represented as

$$\mathbf{v} = \begin{bmatrix} a \\ b \end{bmatrix},$$

where its components on some basis are often arranged as a column of its elements.

In Dirac notation, vectors are represented as *kets* and more often than not, will be decomposed into a linear combination over some basis. A 2-dimensional vector for example, can be written as

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle = \begin{bmatrix} a \\ b \end{bmatrix} \tag{2.1}$$

where $|i\rangle$ is the basis vector of dimension $i$. Usually, the standard (computational) basis is used

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

### 2.1.2 Vector Operations

#### Tensor Product

The tensor product is a fundamental concept in quantum computing and plays a key role in the representation of quantum states and quantum operations. It allows us to extend the notion of a single qubit state to a multi-qubit state and to represent quantum operations that act on multiple qubits simultaneously.

In Dirac notation, the tensor product of two vectors $|\psi_1\rangle$ and $|\psi_2\rangle$ is represented as $|\psi_1\rangle \otimes |\psi_2\rangle$. It is a new vector in a higher-dimensional Hilbert space that describes the joint state of the two vectors. The components of the tensor product vector are given by

$$(|\psi_1\rangle \otimes |\psi_2\rangle)_{i,j} = \psi_{1,i}\psi_{2,j}. \tag{2.2}$$

Here, $\psi_{1,i}$ and $\psi_{2,j}$ are the components of the vectors $|\psi_1\rangle$ and $|\psi_2\rangle$, respectively. The tensor product of two vectors is a useful concept in quantum computing, as it allows us to represent multi-qubit systems as vectors in a higher-dimensional Hilbert space.

#### Conjugate transpose

In classical vector notation the transpose of a vector $\mathbf{v}$ is usually written with its components arranged as rows. To be precise though, since we deal with complex Hilbert spaces in QC, we also need the complex conjugate of that vector

$$\mathbf{v}^\dagger = \begin{bmatrix} x^* & y^* & z^* \end{bmatrix}.$$

In Dirac notation, the same vector would be written as a *bra*

$$\langle v| = \mathbf{v}^\dagger = (\mathbf{v}^*)^T. \tag{2.3}$$

### Inner Product

The inner product of two vectors $|\psi\rangle = \sum_i \psi_i |i\rangle$ and $|\phi\rangle = \sum_i \phi_i |i\rangle$ is written as a *bra-ket* combination

$$\langle\psi|\phi\rangle = \sum_i \psi_i^* \phi_i = \langle\phi|\psi\rangle^* \tag{2.4}$$

### Outer Product

In a similar fashion, the outer product of the said vectors is a *ketbra*

$$|\psi\rangle\langle\phi| = \sum_i \sum_j \psi_i \phi_j^* |i\rangle \langle j| \tag{2.5}$$

## 2.1.3 Density Matrices

In quantum mechanics, a density matrix is a Hermitian, positive semi-definite matrix used to represent the state of a quantum system. The density matrix provides a convenient and complete description of the state of a quantum system, including information about both the probability distribution of the system's wavefunction and the coherence of its quantum superpositions.

A density matrix $\rho$ is defined as

$$\rho = \sum_i p_i |\psi_i\rangle \langle\psi_i| , \tag{2.6}$$

where $p_i$ are the probabilities of the state $|\psi_i\rangle$.

It is important to note that the trace of the density matrix represents the total probability of the system, and it must be equal to 1. That is,

$$\mathrm{Tr}(\rho) = \sum_i \langle\psi_i| \rho |\psi_i\rangle = \sum_i p_i = 1. \tag{2.7}$$

In quantum computing, density matrices are often used to represent mixed states, which are states that cannot be described by a single wavefunction. They are also used to describe the state of a quantum system after it has undergone a measurement, as well as the state of a quantum system when it is in a superposition of multiple states.

### Trace distance

The trace distance is usually used as a metric on the space of *density matrices* and gives a measure of distinguishability between two states. It is the quantum generalization of the Kolmogorov distance for classical probability distributions.

The formal definition of trace distance is half the trace norm of the difference of the two input matrices

$$T(\rho,\sigma) := \frac{1}{2}\|\rho-\sigma\|_1 = \frac{1}{2}\,\mathrm{Tr}\Big\{\sqrt{(\rho-\sigma)^\dagger(\rho-\sigma)}\Big\}. \tag{2.8}$$

For the special case where $\rho, \sigma$ are hermitian

$$T(\rho,\sigma) = \frac{1}{2}\,\mathrm{Tr}\Big\{\sqrt{(\rho-\sigma)^2}\Big\} = \frac{1}{2}\sum_i |\lambda_i|, \tag{2.9}$$

where $\lambda_i$ are the eigenvalues of the hermitian matrix $(\rho-\sigma)$.

## 2.2 Quantum Systems

### 2.2.1 Single-qubit System

A single quantum-bit or *qubit* state can be described by a complex 2-dimensional vector

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle \in \mathbb{C}^2. \tag{2.10}$$

under the restriction $|a|^2 + |b|^2 = 1$. All the possible states of a qubit form a 2-dimensional Hilbert space $\mathcal{H}^2$

As seen by eq. (2.10), qubits are a really powerful generalization of classical bits. While classical bits can only take two distinct values (0 or 1), qubits can range over a continuous subspace of $\mathbb{C}^2$ and can simultaneously be in both the zero-state $|0\rangle$ and the one-state $|1\rangle$. This property is known in quantum mechanics as *superposition* and mathematically is described by the linear combination of two orthogonal states.

### 2.2.2 Multi-qubit systems

In quantum computing, a multi-qubit system is a system composed of multiple qubits. The state of a multi-qubit system is described by a multi-qubit state vector, which is a vector in a high-dimensional Hilbert space. The dimension of this space is given by the product of the dimensions of the individual qubits, so for $n$ qubits, the dimension of the Hilbert space is $2^n$

#### From smaller systems

As an example, let us consider two single-qubit states $|\psi\rangle, |\phi\rangle \in \mathcal{H}^2$. A 2-qubit system can be constructed by the tensor product of these two states

$$|w\rangle = |\psi\rangle \otimes |\phi\rangle = |\psi\rangle\,|\phi\rangle = \begin{bmatrix}\psi_0\\ \psi_1\end{bmatrix} \otimes \begin{bmatrix}\phi_0\\ \phi_1\end{bmatrix} = \begin{bmatrix}\psi_0\phi_0\\ \psi_0\phi_1\\ \psi_1\phi_0\\ \psi_1\phi_1\end{bmatrix} \in \mathcal{H}^4. \tag{2.11}$$

This multi-qubit state $|w\rangle$ can also be written as linear combination of basis states

$$|w\rangle = w_{00}|00\rangle + w_{01}|01\rangle + w_{10}|10\rangle + w_{11}|11\rangle\,, \tag{2.12}$$

where $w_{ij} = \psi_i\phi_j$ and $\langle w|w\rangle = 1$

Though tensor products are fundamental to construct multi-qubit systems, one cannot span a whole vector space with a single product, since they don't introduce correlation. The tensor product of 2 qubit states does indeed construct a multi-qubit system, but the respective single-qubit subsystems remain statistically independent. Ultimately, one needs the ability to correlate two quantum states, which is only achieved through entaglement.

In general, an $n$-qubit state can be written as

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \psi_i |i\rangle\,, \tag{2.13}$$

under the normalization condition $\langle\psi|\psi\rangle = \sum_{i=0}^{2^n-1} |\psi_i|^2 = 1$. The integer notation for basis-vectors can be used interchangeably with the corresponding binary one e.g. $|5\rangle$ instead of $|101\rangle$ and vice versa.

### Entanglement

Quantum entanglement is a phenomenon in quantum mechanics where the quantum states of two or more quantum systems are correlated in a way that cannot be explained by classical mechanics. In quantum computing, entanglement plays a crucial role in many quantum algorithms and protocols.

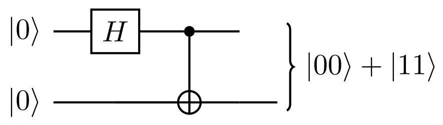


**Figure 2.1.** *Quantum Circuit realizing the maximally-entangled Bell state. Both qubits start from zero state. The Hadamard gate is applied on the first qubit followed by a controlled-NOT gate with the second qubit as target. Controlled operations introduce entanglement.*

One of the simplest examples of entanglement is the Bell state. The Bell state is a two-qubit state that is maximally entangled and can be represented by the following equation in Dirac notation:

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{2.14}$$

Here, $|00\rangle$ and $|11\rangle$ represent the states of the two qubits, and $\frac{1}{\sqrt{2}}$ is a normalization factor. The important feature of the Bell state is that the state of one qubit cannot be described independently of the state of the other qubit i.e. cannot be written as a single tensor product. This is a consequence of entanglement.

### 2.2.3 Quantum Gates

Quantum gates are the basic building blocks of quantum circuits and serve as the analogues of classical logic gates in classical computing. Quantum gates are unitary transformations that manipulate the state of a quantum system and enable the implementation of quantum algorithms.

A quantum gate is represented by a unitary matrix $U$, which acts on the state of a quantum system described by the ket vector $|\psi\rangle$. The action of the gate is given by

$$|\psi'\rangle = U\,|\psi\rangle\,. \tag{2.15}$$

Common examples of single-qubit quantum gates include the Pauli $X$, $Y$, and $Z$ gates, the Hadamard gate $H$, the Phase gate $P$. An example of a single-qubit circuit is illustrated in 2.2, where, starting from the zero-ket state, a Hadamard gate is applied, followed by a Pauli $Y$-gate and finally a measurement. The measurement is the only non-unitary operation in quantum computing which results with the qubit in either state $|0\rangle$ or $|1\rangle$. There are also multi-qubit, controlled-gates that introduce correlation, or *entanglement* in quantum-mechanical sense, such as the Controlled-NOT gate $CX$ and the Toffoli gate $CCX$. These gates, among others, can be combined to form more complex quantum circuits that can perform a variety of quantum operations, including state preparation, quantum state manipulation and quantum error correction. An example of a multi-qubit, controlled gate is illustrated in in the circuit of Fig. 2.3, where the first qubit acts as control-qubit and the rest $n$ qubits as targets.

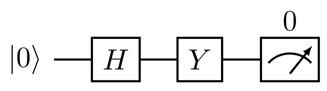


**Figure 2.2.** *Single-qubit circuit with initial state $|0\rangle$. A Hadamard gate is applied to the qubit, followed by a Pauli-Y gate. The resulting state is $|\psi\rangle = YH\,|0\rangle = -i/\sqrt{2}(|0\rangle - |1\rangle)$. Finally, a measurement takes place which counts whether the result state is $|0\rangle$. Due to the Hadamard gate, the probability of measuring state $|0\rangle$ is $p(0) = \left|-i/\sqrt{2}\right|^2 = 0.5$*

It is important to note that all quantum gates are reversible, meaning that the inverse of a gate exists and can be used to undo the action of the gate. This is a fundamental property of quantum gates and is in contrast to classical gates, which are typically irreversible. Additionally, as mentioned, quantum gates must be unitary, meaning that they preserve the normalization of the state vectors they operate on, and they must be efficient to implement in a practical quantum computing setting. These two properties can be expressed with the following restriction

$$U^\dagger U = \mathbb{1} \tag{2.16}$$

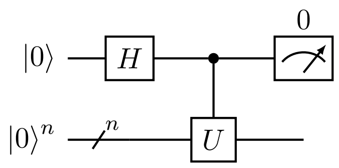


**Figure 2.3.** *Multi-qubit circuit with a controlled operation $CU$. $U$ operator is applied in the second $n$-qubit register only when the control (first from above) qubit has a $|1\rangle$ component.*

## 2.3 Useful Quantum Circuits for this work

### 2.3.1 Inner Product as circuit

Given two quantum states $|x\rangle, |y\rangle$, the goal is to compute the product $\langle x|y\rangle$ . Starting from the state

$$|\psi_0\rangle = \frac{1}{\sqrt{2}}(|0\rangle|x\rangle + e^{i\phi}|1\rangle|y\rangle) \tag{2.17}$$

we apply a Hadamard gate on the first ancilla qubit

$$\begin{aligned}|\psi\rangle =&(H \otimes I)|\psi_0\rangle \\ =&\frac{1}{2}(|+\rangle|x\rangle + e^{i\phi}|-\rangle|y\rangle) \\ =&\frac{1}{2}\Big(|0\rangle(|x\rangle + e^{i\phi}|y\rangle) + |1\rangle(|x\rangle - e^{i\phi}|y\rangle)\Big)\end{aligned}$$

The probability of measuring 0 is

$$\begin{aligned}p(0) =& \langle\psi|(|0\rangle\langle 0| \otimes I)|\psi\rangle \\ =&\frac{1}{4}\Big(\langle x|x\rangle + \langle y|y\rangle + e^{i\phi}\langle x|y\rangle + e^{-i\phi}\langle y|x\rangle\Big) \\ =&\frac{1}{4}\Big(2 + 2\,\mathrm{Re}\Big\{e^{i\phi}\langle x|y\rangle\Big\}\Big)\end{aligned}$$

and finally

$$\mathrm{Re}\Big\{e^{i\phi}\langle x|y\rangle\Big\} = 1 - 2p(0) \tag{2.18}$$

Since $\langle x|y\rangle$ is in general a complex number we want to estimate both its real and imaginary

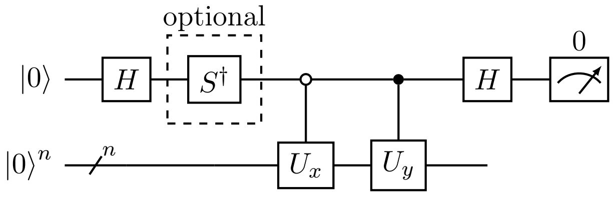


**Figure 2.4.** *Circuit for Inner product. The $S^\dagger$ gate is added to the circuit when the imaginary part of the inner product is needed. The circuits $U_x, U_y$ evolve $|0\rangle^n$ to the states $|x\rangle, |y\rangle$ respectively. The white-coloured control operation implies conditional on the $|0\rangle$ state, instead of $|1\rangle$.*

parts. This is achieved by proper selection of the initial phase parameter in 2.17. By setting $\phi = 0$ in we get $\operatorname{Re}\langle x|y\rangle$ from 2.18. To get $\operatorname{Im}\langle x|y\rangle$ we set $\phi = -\frac{\pi}{2}$ respectively.

### 2.3.2 The Hadamard Test

The Hadamard test is a special case of the inner product computational scheme, which computes the expected value $\langle\psi| U |\psi\rangle$.

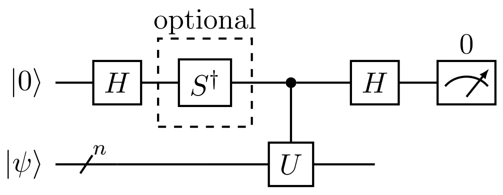


**Figure 2.5.** *Circuit for Hadamard Test. The $S^\dagger$ gate is added to the circuit when the imaginary part of the expectation value is needed.*

### 2.3.3 Quantum Amplitude Estimation

Given an operator $A$ dividing a system in two subspaces

$$A|0\rangle = \sqrt{1-\alpha}\,|\psi_0\rangle + \sqrt{\alpha}\,|\psi_1\rangle$$

amplitude estimation algorithms form a family of methods that estimate the amplitude of the "good" state $\alpha = |\langle\psi_1|\psi_1\rangle|^2$.

This task has first been investigated by Brassard et al. [30] in 2000 and their algorithm uses a combination of the Grover operator and Phase estimation.

**Theorem 2.1** (Amplitude Estimation [30]). *There is a quantum algorithm called "Amplitude Estimation" which for any integer $k > 0$ takes as input a 2-state quantum system $|\psi\rangle$ and forms a unitary transformation $Q = AS_0A^\dagger S_{\psi_1}$, which is applied $j$ times. The algorithm outputs $0 \le \tilde{\alpha} \le 1$, an estimate of $\alpha = \langle\psi_1|\psi_1\rangle$, such that*

$$|\tilde{\alpha} - \alpha| \le 2k\pi\frac{\sqrt{a(1-a)}}{M} + \pi^2\frac{k^2}{M^2}$$

- *with success probability at least $8/\pi^2$ when $k = 1$*
- *with probability greater than $1 - \frac{1}{2(k-1)}$ for $k \ge 2$.*

*$M$ is a dimension hyperparameter that controls precision.*

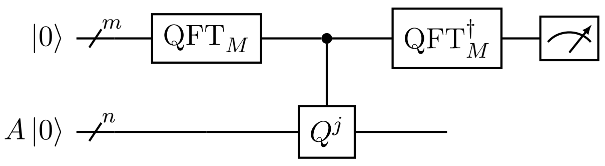


**Figure 2.6.** *Circuit for original Amplitude Estimation.*

## 2.4 The Quantum Linear System Problem

Linear systems of equations need no introduction. They are used excessively in most areas of science and technology, including machine learning, solving differential equations, computer graphics, signal processing and countless other engineering applications. Being a subject of such importance, solving linear has been the cutting edge of computational science even before the invention of classical computers. Thus, it came as no surprise that as quantum computing hardware was — and still is — at its infancy, many quantum algorithms for linear systems have already emerged.

It is known, that solving an $N \times N$ Linear System Problem (LSP) with a classical computer scales polynomially in $N$. In contrast, as Harrow-Hassidim-Lloyd (HHL) showed [9], a quantum algorithm scales logarithmically in $N$, suggesting that quantum computers can provide an exponential speedup for certain linear system problems. As awe-inspiring as that may sound at first, there is a number of limitations that must be taken int account. First and foremost, all the algorithms of the HHL family - and most others- try to address the Quantum Linear System Problem (QLSP), where the goal is to prepare a quantum state $|x\rangle$ that is proportional to a vector $\mathbf{x}$ that satisfies the original LSP.

Specifically, given the LSP

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{2.19}$$

where $A$ is an $N \times N$ sparse hermitian matrix with condition number $\kappa$, $\mathbf{b}$ a sparse vector and $x$ the solution of the system, the QLSP is the respective system

$$A\,|x\rangle = |b\rangle\,, \tag{2.20}$$

where $A$ is a normalized version of $\mathbf{A}$ with spectral norm $\|A\| \leq 1$ and $|b\rangle = \frac{\mathbf{b}}{\|\mathbf{b}\|}$ the normalized right-hand side vector. We must strongly emphasize at this point that the LSP and its QLSP counterpart are two distinct problems w.r.t their result and because the latter does *not* offer direct access to the values of $|x\rangle$ with better complexity, but rather it's assumed that the user is interested in applying an operator $M$ to the solution. Thus, the measurable result of the algorithm can only be a quantity $\langle x|\,M\,|x\rangle$ and not $|x\rangle$ itself.

If both $\mathbf{A}$ and $\mathbf{b}$ are sparse the original HHL solves for $|x\rangle$ in $O(\kappa^2 \log N/\epsilon)$ time. For comparison, the most well-established classical algorithm the Conjugate-Gradient method runs in $O(Nsk)$, where $s$ is the sparsity of $\mathbf{A}$. The latter though gives the user direct access to the solution vector, something that quantum algorithms cannot provide without sacrificing their speedup. So, one should compare quantum linear system algorithms with classical ones that also return summaries of the solution vector $\mathbf{x}^\dagger\mathbf{M}\mathbf{x}$, in which case a time of $O(N\sqrt{k})$ can be achieved.

Chapter 3

# The Variational Quantum Linear Solver

## 3.1 Summary

The Variational Quantum Linear Solver (VQLS) [1] is a Variational Hybrid Quantum-Classical (VHQC) algorithm for linear systems of equations and specifically for the solution of the Quantum Linear System Problem (QLSP), designed with NISQ devices in mind. Given the LSP $\mathbf{Ax} = \mathbf{b}$ and the equivalent QLSP $A\left|x\right\rangle = \left|b\right\rangle$, the quantum part of the algorithm uses a cost function which captures the similarity between vectors $A\left|x\right\rangle$ and $\left|b\right\rangle$, by means of either a local or global hamiltonian. Classical optimization schemes can be then utilized in order to minimize and finally terminate when a desired precision $\epsilon$ is reached.

Although VQLS is heuristic and complexity analysis is considered quite challenging, the authors provided numerical simulations indicating efficient scaling in $\kappa, \epsilon$ and $N$. Specifically, they found evidence of (at worst) linear scaling in $\kappa$, logarithmic scaling in $1/\epsilon$ and polylogarithmic scaling in $N$ [1].

## 3.2 Overview

Figure fig. 3.1 shows a descriptive overview of the VQLS algorithm. The input to VQLS is

- an efficient gate sequence $U$ that prepares the quantum state $\left|b\right\rangle = U\left|0\right\rangle$ which is proportional to the right-hand side vector $\mathbf{b}$
- a linear decomposition of the normalized matrix $A$ into unitaries of the form

$$A = \sum_{l=1}^{L} c_l A_l, \tag{3.1}$$

- A vector of parameters $\boldsymbol{\alpha}$

where $A_l$ are unitary matrices and $c_l$ complex numbers. Intuitively, this linear decomposition into unitaries is essential to construct the required circuits since all quantum gates are unitary operators themselves. Here, the following assumptions are made:

1. $L$ is only a polynomial function of the number of qubits, $n$
2. The QLSP matrix has finite condition number $\kappa < \infty$ and its spectral norm satisfies $\|A\| \leq 1$
3. $A_l$ unitaries can be implemented with efficient quantum circuits.

Information over efficient decomposition method for sparse $A$ can be found at [1].

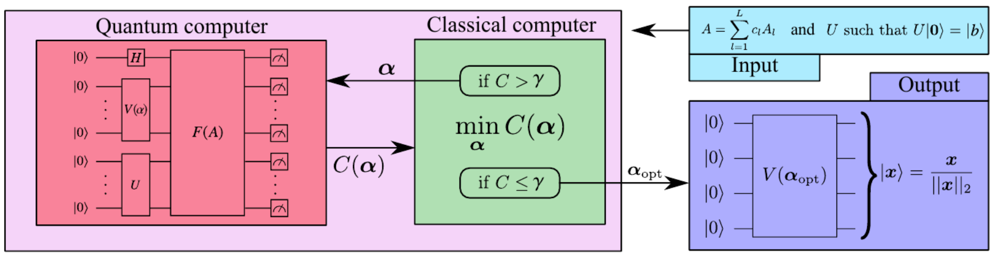

Quantum computer
$|0\rangle$
$H$
$V(\alpha)$
$F(A)$
$U$
Classical computer
$\boldsymbol{\alpha}$
if $C > \gamma$
$\min_{\boldsymbol{\alpha}} C(\boldsymbol{\alpha})$
$C(\boldsymbol{\alpha})$
if $C \leq \gamma$
$\boldsymbol{\alpha}_{\text{opt}}$
$A = \sum_{l=1}^{L} c_l A_l$ and $U$ such that $U|\mathbf{0}\rangle = |b\rangle$
Input
Output
$V(\boldsymbol{\alpha}_{\text{opt}})$
$|\boldsymbol{x}\rangle = \frac{\boldsymbol{x}}{||\boldsymbol{x}||_2}$

**Figure 3.1.** *Schematic diagram of VQLS algorithm from [1]. The input to VQLS is a matrix $A$ written as a linear combination of unitaries $A_l$ and a short-depth quantum circuit $U$ which prepares the state $|b\rangle$. The output of VQLS is a quantum state $|x\rangle$ that is approximately proportional to the solution of the linear system $Ax = b$. Parameters $\alpha$ in the ansatz $V(\alpha)$ are adjusted in a hybrid quantum-classical optimization loop until the cost $C(\alpha)$ (local or global) is below a user-specified threshold. When this loop terminates, the resulting gate sequence $V(\alpha_{opt})$ prepares the state $|x\rangle = x/\|x\|_2$, from which observable quantities can be computed. Furthermore, the final value of the cost $C(\alpha_{opt})$ provides an upper bound on the deviation of observables measured on $|x\rangle$ from observables measured on the exact solution.*

As mentioned, in order to solve the QLSP one must prepare a state $|x\rangle$ s.t. $A\,|x\rangle$ is as close as possible to $|b\rangle$. In order to achieve that, VQLS employs an ansatz $V(\boldsymbol{\alpha})$ that prepares a candidate solution $|x(\boldsymbol{\alpha})\rangle = V(\boldsymbol{\alpha})\,|0\rangle$. Parameters $\boldsymbol{\alpha}$ are given as input to the quantum computer, which prepares $|x(\boldsymbol{\alpha})\rangle$ and runs an efficient quantum circuit that estimates a cost function $C(\boldsymbol{\alpha})$. The value of the cost function quantifies how much the proposed state $A\,|x\rangle$ differs from the target state $|b\rangle$. This value $C(\boldsymbol{\alpha})$ is then returned to the classical computer which adjusts parameters $\boldsymbol{\alpha}$ according to a classical optimization algorithm, attempting to reduce the cost. The process is repeated until a termination condition of the form $C(\boldsymbol{\alpha}) \leq \gamma$ is achieved, at which point VQLS outputs the solution parameters $\boldsymbol{\alpha}_{\text{opt}}$.

Finally, $\boldsymbol{\alpha}_{\text{opt}}$ can be used to prepare the solutions state $|x(\boldsymbol{\alpha}_{\text{opt}})\rangle = V(\boldsymbol{\alpha}_{\text{opt}})\,|0\rangle$. Once the solution state is achieved we can finally measure quantities of interest in order to extract valuable information for the solution vector. The deviation of observable expectation values from those of the exact solution can be upper bounded based on the cost function. Thus, one can beforehand determine a desired error tolerance $\epsilon$, where

$$\epsilon = \frac{1}{2}\|\,|x_0\rangle\langle x_0| - |x(\boldsymbol{\alpha}_{\text{opt}}\rangle\langle x(\boldsymbol{\alpha}_{\text{opt}}|\,\|_1 = \frac{1}{2}\,\mathrm{Tr}\left\{\sqrt{(|x_0\rangle\langle x_0| - |x(\boldsymbol{\alpha}_{\text{opt}}\rangle\langle x(\boldsymbol{\alpha}_{\text{opt}}|)^2}\right\} \tag{3.2}$$

is the trace distance between the exact solution $|x_0\rangle$ and the approximate solution $|x(\boldsymbol{\alpha}_{\text{opt}})\rangle$.

This $\epsilon$ corresponds to a threshold value $\gamma$ that the final cost $C(\boldsymbol{\alpha})$ must achieve.

## 3.3 Cost functions

At this point, we shall see how according to [1] the cost functions can be described as expectation values of global or local Hamiltonians. To avoid cumbersome notation, we will thereafter write $|x(\boldsymbol{\alpha})\rangle$ as $|x\rangle$ or more often use $|\psi\rangle = A|x\rangle$. The simplest cost function can be constructed by taking the overlap of the projector $|\psi\rangle\langle\psi|$ with the subspace orthogonal to vector $|b\rangle$:

$$\hat{C}_G = \mathrm{Tr}\{|\psi\rangle\langle\psi| (\mathbb{1} - |b\rangle\langle b|)\} = \langle\psi|H_G|\psi\rangle \tag{3.3}$$

The second way to view this cost function is the expectation value of an effective Hamiltonian

$$H_G = \mathbb{1} - |b\rangle\langle b| \tag{3.4}$$

which is similar to the final Hamiltonian in [5]. The $\hat{C}_G$ function is small if $|\psi\rangle$ nearly proportional to $|b\rangle$ or if the norm of $|\psi\rangle$ itself is small. The latter is not desired, so to combat this, one can normalize by

$$C_G = \frac{\hat{C}_G}{||\psi||^2} = \frac{\langle\psi|H_G|\psi\rangle}{\langle\psi|\psi\rangle} = 1 - ||\langle b|\Psi\rangle||^2, \tag{3.5}$$

where $|\Psi\rangle = \frac{|\psi\rangle}{\sqrt{\langle\psi|\psi\rangle}}$ is a normalized state.

As it is noted in the original article [1], global functions such as eq. (3.3) and eq. (3.5) may introduce barren plateaus in the energy field, which leads to exponentially vanishing gradients with respect to the number of qubits $n$ [31]. For improved trainability as $n$ grows larger, the authors introduce local versions of the aforementioned costs

$$\hat{C}_L = \langle\psi|H_L|\psi\rangle \tag{3.6}$$

$$C_L = \frac{\hat{C}_L}{\langle\psi|\psi\rangle}, \tag{3.7}$$

where the effective Hamiltonian is

$$H_L = U\left(\mathbb{1} - \frac{1}{n}\sum_{j=1}^{n} |0_j\rangle\langle 0_j| \otimes \mathbb{1}_{\bar{j}}\right)U^\dagger, \tag{3.8}$$

where $|0_j\rangle$ is the zero state on qubit $j$ and $\mathbb{1}_{\bar{j}}$ the identity on all qubits but qubit $j$. It is also shown that the following inequalities apply

$$\hat{C}_L \leq \hat{C}_G \leq n\hat{C}_L \tag{3.9}$$

$$C_L \leq C_G \leq nC_L, \tag{3.10}$$

from which follows by the sandwich-theorem that $C_L = 0 \iff C_G = 0$ and $\hat{C}_L = 0 \iff$

$\hat{C}_G = 0$.

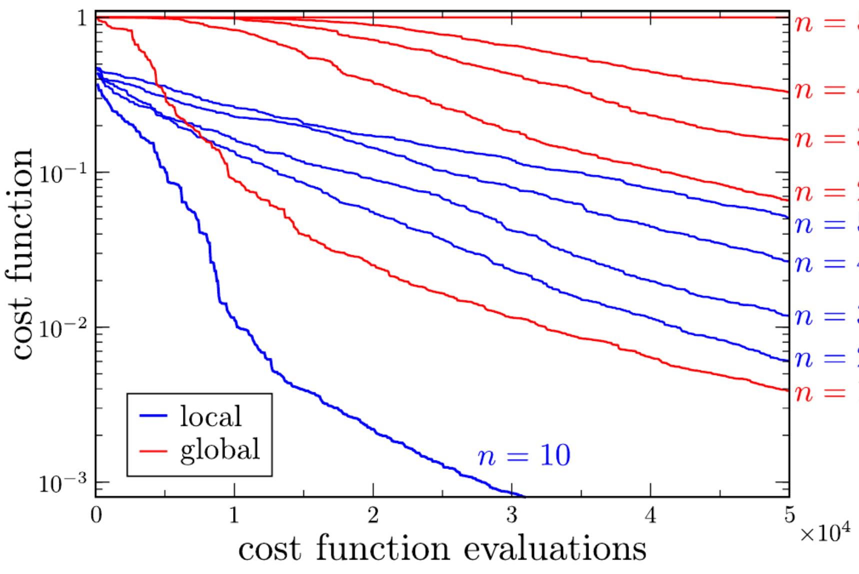


**Figure 3.2.** *Comparison of local $C_L$ and global $C_G$ cost performance from [1]. In all cases $\kappa = 20$. For each $n \in \{10, ..., 50\}$, the cost value versus the number of cost function evaluations is plotted. As n increases it becomes increasingly hard to train to global cost function. At $n = 50$, the optimization cannot significantly lower $C_G$ below a value of one. On the other hand, $C_L$ is trainable for all values of n considered.*

As one sees in fig. 3.2, as the number of qubits $n$ increases the global cost function $C_G$ becomes harder to optimize. Local cost $C_G$, on the other hand, remains trainable no matter how much $n$ increases. It is hence recommended by the authors [1] that local cost functions are used, at least for large-scale implementations.

### 3.3.1 Cost function bounds

As shown in [1], the following bounds hold for the cost functions:

$$\hat{C}_G \geq \frac{\epsilon^2}{\kappa^2}, \qquad C_G \geq \frac{\epsilon^2}{\kappa^2 \langle\psi|\psi\rangle}, \tag{3.11}$$

$$\hat{C}_L \geq \frac{\epsilon^2}{n\kappa^2}, \qquad C_L \geq \frac{\epsilon^2}{n\kappa^2 \langle\psi|\psi\rangle}. \tag{3.12}$$

These are useful because they can be used to define termination conditions for VQLS with respect to the desired precision $\epsilon$. The right-hand sides of eq. (3.11), eq. (3.12) can be used as the $\gamma$ quantity shown in fig. 3.1. Note that $\langle\psi|\psi\rangle \leq 1$, hence the $\langle\psi|\psi\rangle$ terms in the denominators of the normalized cost function inequalities can be omitted, if one needs less strict bounds. During execution though, since these terms are already computed, there is no need to omit these terms.

Finally, by employing the operational meaning of the trace distance [28] it can be shown that for any POVM element M,

$$\epsilon \geq D(M), \tag{3.13}$$

where

$$D(M) = |\langle x|M|x\rangle - \langle x_0|M|x_0\rangle| \tag{3.14}$$

measures the difference of expectation values with respect to $|x\rangle$ and $|x_0\rangle$. If $M$ is in the general case any Hermitian observable, eq. (3.13) becomes

$$\epsilon \geq \frac{D(M)}{2|M|} \tag{3.15}$$

and thus eq. (3.11) and eq. (3.12) are bounded by *observable* values.

## 3.4 Cost evaluation

In theory, all the aforementioned cost functions can be efficiently evaluated by the Hadamard Test[1] and then applying classical post-processing. The simplest term is the evaluation of $\langle\psi|\psi\rangle$ which, by considering the linear decomposition of the system matrix $A = \sum_l c_l A_l$, can be written as

$$\langle\psi|\psi\rangle = \sum_{ll'}^{L} c_l c_{l'}^* \beta_{ll'} \tag{3.16}$$

where

$$\beta_{ll'} = \langle\mathbf{0}|\, V^\dagger A_{l'}^\dagger A_l V \,|\mathbf{0}\rangle \tag{3.17}$$

There are $L(L-1)/2$ different terms in eq. (3.17) which can be estimated with the Hadamard Test. This requires acting with $V$ on $|\mathbf{0}\rangle$ and then using an ancilla as the control qubit, applying $CA_l$ and then $CA_{l'}^\dagger$, where $CW$ stands for controlled-$W$ operation.

If one wants to evaluate the global cost functions $\hat{C}_G, C_G$, they will also need

$$|\langle b|\psi\rangle|^2 = \left|\langle\mathbf{0}|\, U^\dagger A V \,|\mathbf{0}\rangle\right|^2 = \sum_{ll'} c_l c_{l'}^* \gamma_{ll'}, \tag{3.18}$$

where

$$\gamma_{ll'} = \langle\mathbf{0}|\, V^\dagger A_{l'}^\dagger U \,|\mathbf{0}\rangle \langle\mathbf{0}|\, U^\dagger A_l V \,|\mathbf{0}\rangle\,. \tag{3.19}$$

The $\gamma_{ll}$ are estimated by applying $U^\dagger A_l V$ to $|\mathbf{0}\rangle$ and then measuring the probability of $|\mathbf{0}\rangle$ state. The rest $L(L-1)/2$ terms where $l \neq l'$ can be estimated with a Hadamard Test. In [1], though, a novel circuit, the Hadamard-Overlap Test, is introduced for that purpose which eliminates the need to control all the unitaries, at the expense of doubling the number of qubits.

Finally, for $\hat{C}_L, C_L$, one needs to estimate the terms

$$\delta_{ll'}^j = \langle\mathbf{0}|\, V^\dagger A_{l'}^\dagger U (|0_j\rangle\langle 0_j| \otimes \mathbb{1}_{\bar{j}}) U^\dagger A_l V \,|\mathbf{0}\rangle\,, \tag{3.20}$$

where the global $|\mathbf{0}\rangle\langle\mathbf{0}|$ of eq. (3.19) has been substituted with its correspondent local operator $|0_j\rangle\langle 0_j| \otimes \mathbb{1}_{\bar{j}}$. The $j$ subscript implies application of its operator to the $j$-th qubit,

[1]see section 2.3.2

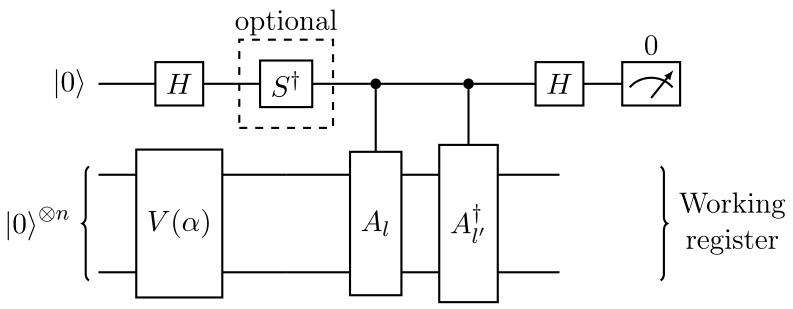


**Figure 3.3.** *Hadamard Test used to compute coefficients $\beta_{ll'} = \langle \mathbf{0}| V^\dagger A_{l'}^\dagger A_l V |\mathbf{0}\rangle$ [1]. The $S^\dagger$ gate is added to the circuit when the imaginary part of the expectation value is needed.*

while $\bar{j}$ implies application to every other qubit but $j$.

By exploiting the fact that $|0_j\rangle\langle 0_j| = \frac{1}{2}(Z_j + I_j)$, eq. (3.20) can be written as

$$\delta_{ll'}^j = \frac{1}{2}\left(\langle \mathbf{0}| V^\dagger A_{l'}^\dagger U \left((I_j + Z_j) \otimes I_{\bar{j}}\right) U^\dagger A_l V |\mathbf{0}\rangle\right) = \frac{1}{2}(\beta_{ll'} + \zeta_{ll'}^j), \tag{3.21}$$

where

$$\zeta_{ll'}^j = \langle \mathbf{0}| V^\dagger A_{l'}^\dagger U \left(Z_j \otimes I_{\bar{j}}\right) U^\dagger A_l V |\mathbf{0}\rangle \tag{3.22}$$

This final version of eq. (3.20) is used to construct the circuit which estimates $\delta_{ll'}^j$.

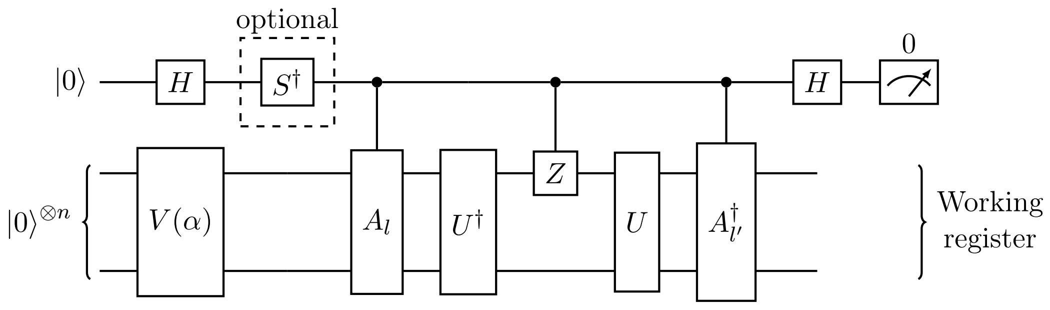


**Figure 3.4.** *Hadamard Test used to compute $\zeta_{ll'}^i$ coefficients as indicated in eq. (3.22) [1]. Shown is the case for $j = 1$, when controlled-Z gate is applied to first qubit of working register. The $S^\dagger$ gate is added to the circuit when the imaginary part of the expectation value is needed.*

## 3.5 Ansatz

VQLS works by preparing the state $|x\rangle = V(\boldsymbol{\alpha})|\mathbf{0}\rangle$. The unitary operator $V(\boldsymbol{\alpha})$ consists of a sequence of gates with trainable continuous parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_L)$ and can in general be expressed as

$$V(\boldsymbol{\alpha}) = G_{k_L}(\alpha_L) \ldots G_{k_i}(\alpha_i) \ldots G_{k_1}(\alpha_1). \tag{3.23}$$

Quantum ansatzes are mostly empirical sequences of operations, quite similar to feed-forward neural networks. The discrete parameters $\mathbf{k} = (k_L, \ldots, k_1)$ encode the types of gates and their placement in the circuit i.e. the ansatz architecture. If $k$ is constant, we are referring to "fixed-structure" ansatz, since one optimizes only over $\boldsymbol{\alpha}$. Although less versatile, fixed ansatzes are simpler than "variable-structure" ones, where one optimizes with respect to $\mathbf{k}$ as well. Lastly, another type that can be employed is the Quantum Alternating Operator Ansatz (QAOA) [32, 33], which is known to be universal as the number of its layers tend to infinity [34, 35].

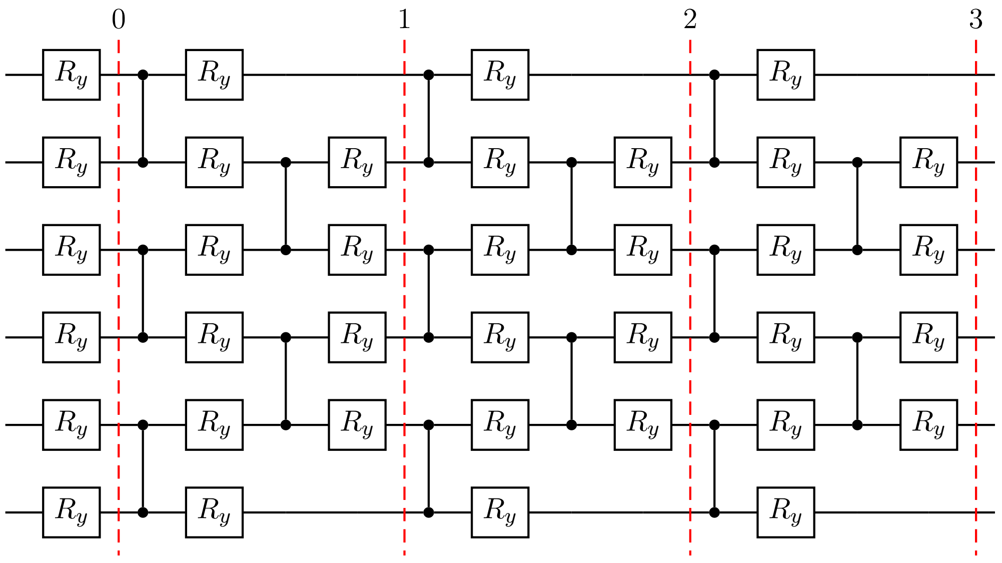


**Figure 3.5.** *Fixed-structure layered Ansatz for $V(\mathbf{a})$ [1]. Each layer introduces $R_y(a_i)$-rotations (exploration) and controlled-Z operations acting on alternating pairs of neighbouring qubits (correlation). If the linear system is restricted to real numbers, $R_y$ gates are sufficient, since they don't produce complex output. For fixed layer depth the number of variational parameters scales as $O(n)$. The example shows an ansatz with $n = 6$ qubits (linear system with size $2^6 \times 2^6$) and three layers.*

When dealing with real QPUs, it is wise to select hardware-efficient Ansatzes [36] using a native gate set $\mathrm{G} = \{G_k(\alpha)\}$ which also takes account the processor's native topology.

## 3.6 Machine Learning Enhanced VQLS for parametric systems

As we mentioned earlier, VQLS is a variational quantum algorithm, which solves a QLSP by minimizing a Hamiltonian loss function with the help of a quantum ansatz. The optimized circuit parameters are then given as input to the ansatz which stores the solution of the linear system in a quantum state. This state is finally used to extract quantities of interest, usually in the form of an expectation operator. While that holds for the case of a single QLSP, it is common that one needs to solve a series of correlated linear systems generated by a distribution of physical parameters[2]. This is often the case in stochastic analysis and optimization problems, where a multi-query exploration of the solution space is required.

In a multitude of applications, such as uncertainty propagation, parameter inference, op-

[2]not to be confused with VQLS ansatz parameters.

timization, sensitivity analysis etc. it is required to produce sufficient number of samples or solution datasets. These, in turn, require the solution of parametric linear systems, and besides the often unavoidable, immense matrix sizes, one has to deal with an overwhelming number of LSP instances which often exceed available computational resources. While quantum algorithms, and specifically VQLS, are expected to deal with the former challenge by exponentially reducing the computational complexity of individual LSPs, there still remains space for further acceleration of the overall parametric problem.

In this regard, it is essential to derive efficient methods that exploit the continuity of the LSP solution space with the respect to the parameters that produce the linear system, hereafter called physical parameters $\mathbf{p} \in \mathbb{R}^d$. Starting from the linear system with equation $\mathbf{A}(\mathbf{p}) \cdot \mathbf{x} = \mathbf{b}$, with $\mathbf{A}(\mathbf{p})$ being the parametrized system matrix, $\mathbf{x}$ the solution sought and $\mathbf{b}$ the right-hand side vector. The equivalent QLSP is the system with equation $A(\mathbf{p})\,|x\rangle = |b\rangle$. Finally, when a variational quantum algorithm such as VQLS is used as solver, the system takes the form

$$A(\mathbf{p})\,|x(\boldsymbol{\alpha})\rangle = |b\rangle\,, \tag{3.24}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_r) \in \mathbb{R}^r$ the VQLS real parameter-vector and $|x(\boldsymbol{\alpha})\rangle$ the proposed solution of the QLSP. Using the cost function from eq. 3.7 the optimal parameters are obtained by the following minimization problem

$$\boldsymbol{\alpha}_{\text{opt}} = \arg\min_{\boldsymbol{\alpha}} C(\mathbf{p}, \boldsymbol{\alpha}) = \arg\min_{\boldsymbol{\alpha}} \langle \Psi(\mathbf{p}, \boldsymbol{\alpha}) | H_L | \Psi(\mathbf{p}, \boldsymbol{\alpha}) \rangle\,, \tag{3.25}$$

It is evident from eq. 3.25 that for every matrix $A(\mathbf{p})$, a search over parametric space $\mathbb{R}^r$ is required in order to reach the solution $|x(\boldsymbol{\alpha}_{\text{opt}})\rangle$ which minimizes the Hamiltonian cost $C_L$.

Our proposed scheme uses a machine-learning model $\mathcal{M} : \mathbb{R}^d \to \mathbb{R}^r$ in order to propose refined initial guesses to the VQLS algorithm. Instead of letting the initial VQLS parameters be random or constant, we take advantage of the assumed continuity of the cost function $C(\mathbf{p}, \boldsymbol{\alpha})$, so that for every physical parameter $\mathbf{p}$, a refined initial guess $\boldsymbol{\alpha}_0$ is proposed that is as close as possible to the optimal ones $\|\boldsymbol{\alpha}_\mathbf{0} - \boldsymbol{\alpha}_{\text{opt}}\| \leq \epsilon$ with some probability $P_{\mathcal{M}}$ depending on the chosen model. The acceptable error $\epsilon$ between the proposed and optimal parameters is a hyperparameter for the overall scheme and depends on the value of the training loss.

Since, in general, no training dataset will be available, one has to initially rely on the usual workflow without refined initial parameter proposition i.e. $\boldsymbol{\alpha}_0^i = \mathcal{S}(\mathbf{p}^i)$ for $i = 1, \ldots, N_{\text{train}}$ using some generic strategy $\mathcal{S}$. Once the first optimal solutions are produced, the model can be trained on the respective dataset $(\mathbf{p}^i, \boldsymbol{\alpha}_{\text{opt}}^i)$, $i = 1, \ldots, N_{\text{train}}$. The trained model thereafter proposes refined initial parameters $\boldsymbol{\alpha}_0^i = \mathcal{M}(\mathbf{p}^i)$ for $i = N_{\text{train}} + 1, \ldots, N_{\text{total}}$. Again, $N_{\text{train}}$ is considered a hyperparameter which must be chosen to balance the trade-off between accuracy and training-time.

In the model architecture context, in order to minimize training time without adding to the

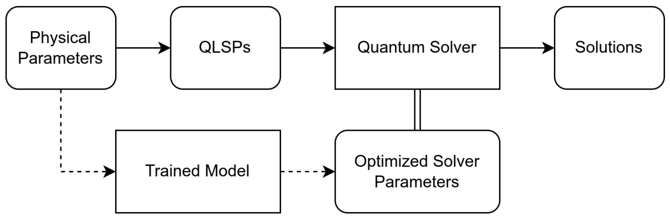


**Figure 3.6.** *Computational workflow for the machine-learning-enhanced VQLS scheme for parametric systems. Initially, in order to produce training data, one executes the process as indicated by the solid-line arrows. Once enough training data have gathered, the machine-learning model is trained with $x^i$ as input and $\alpha^i$ as output. The computational scheme thereafter changes (indicated by the dashed arrows) as the model now proposes initial VQLS parameters $\alpha^i$, which are closer to the optimal, thus reducing the optimization cost of VQLS.*

overall execution time, it is desirable to choose a ML model $\mathcal{M}$ with low complexity and efficient scaling with respect to its output. In our scheme, the only hyperparameter that can be controlled in this regard is the complexity of the ansatz $V$ and more specifically the dimension $r$ of the ansatz parameter vector $\boldsymbol{\alpha}$. An ansatz that scales polynomially with the number of qubits is desirable because it enables one to construct models that remain efficient as the size of the system increases. Specifically, in our case $r$ should scale at most as $\mathcal{O}(\text{poly}(n))$, where $n$ the number of qubits required for the system, so that low-complexity, efficient ML models are sustainable. Some known ansatzes with polynomial qubit scaling include the Quantum Approximate Optimization Algorithm (QAOA) ansatz [32] and the Hardware-Efficient ansatz [36].

Chapter 4

# Numerical examples

To evaluate the performance of the proposed scheme we conducted the comparative experiments described in the subsequent section. The quantum circuit simulations were driven by the Qiskit open-source framework and specifically the *statevector simulator* [37]. The matrix generator for the parametric $2^n \times 2^n$ QLSP is chosen to be

$$A(\mathbf{p}) = \frac{1}{\xi}\left(I + \frac{1}{C}\sum_{j,k=1}^{n} c_{ij}(\mathbf{p})(X_j \otimes Z_k)\right), \tag{4.1}$$

where $n$ is the number of effective qubits used in the circuit, $c_{ij}(\mathbf{p})$ are coefficients that depend on the physical parameters $\mathbf{p}$ and $C, \xi$ are normalization constants. In both examples, the physical parameters $\mathbf{p} = (p_0, p_1, p_2)^T$ are uniformly distributed in a 3-dimensional unit cube $D : [0,1] \times [1,2] \times [3,4]$. At each example we compare pure VQLS schemes with ones that fall under our paradigm. Since the circuits are simulated, the comparison is conducted in relation to the required number of iterations to solve each instance of the parametric system. The parametric QLSP is discretized into a total of $N_{\text{total}}$ regular QLSPs, with equations

$$A(\mathbf{p}^i) \left|x(\boldsymbol{\alpha}^i)\right\rangle = |b\rangle, \tag{4.2}$$

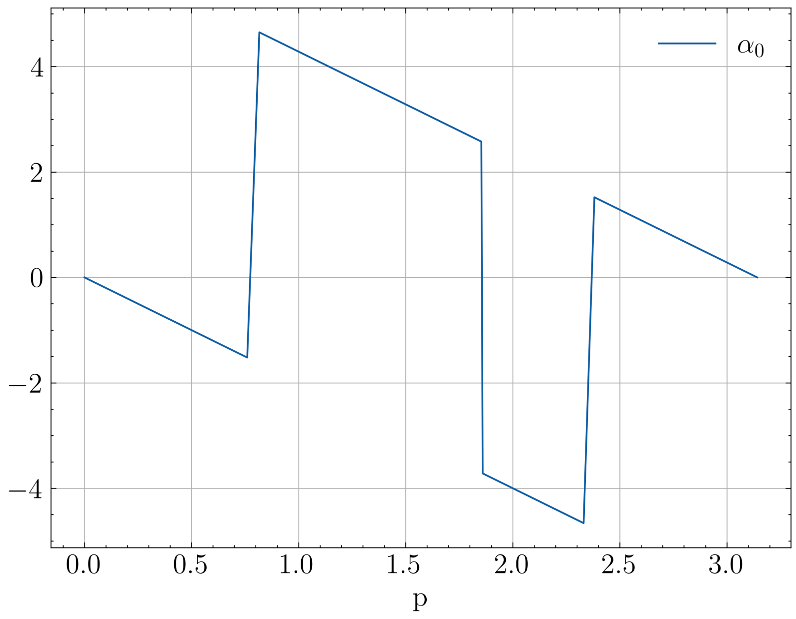
and $i = 1, \ldots, N_{\text{total}}$. For all examples the right-hand side is set to $|b\rangle = |\mathbf{0}\rangle \in \mathbb{C}^{2^n}$ and the total number of solutions $N_{\text{total}} = 1000$.

## 4.1 Convergence and continuity

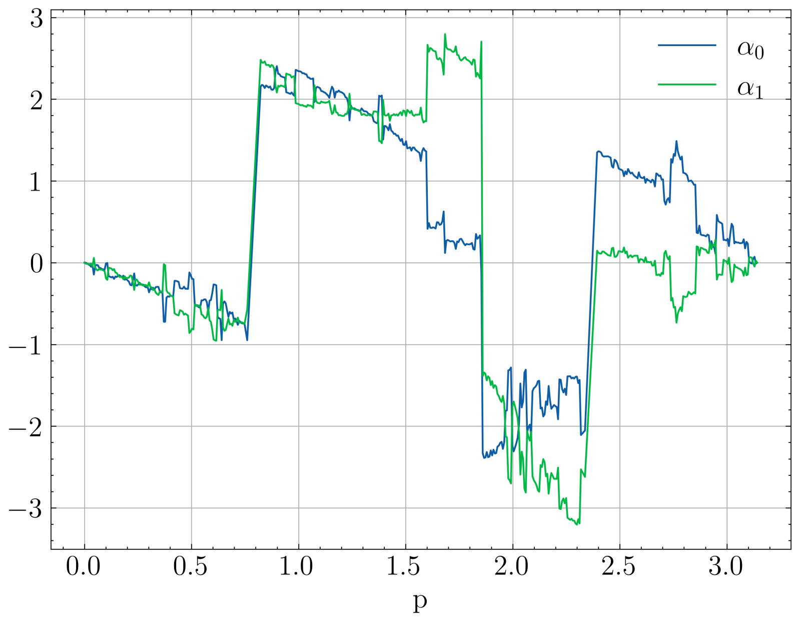
A main problem when using variational algorithms is that the Hamiltonian loss-functions and especially their local versions are inherently non-convex which increases the probability for different optimization paths with neighboring initial parameters to land astray at completely different local-minima. Furthermore, quantum ansatzes are inherently periodic, since most parametric quantum gates themselves are not bijective with respect to their parameters. In fact, quantum gates are essentially rotations with period of at most $4\pi$ and the parameters themselves are rather angles of rotation in some complex Hilbert space. Thus, when an ansatz is deployed, in general there is often a multitude of parameter values that can produce the same state. Besides periodicity, an ansatz contains many different paths through which the quantum states are prepared, resulting in different quantum states that may possess the same properties and more specifically the same probability distribution when measured under a specific basis. Lastly, when dealing with real quantum processors, the unavoidable hardware noise will most definitely propagate through the optimization path resulting again in discontinuous optimal values.

Another source of discontinuities which may arise in low-qubit systems is over-parametrization of the quantum ansatz. The required number of parameters to fully span an $n$-qubit system's Hilbert space is $N_{\text{span}} = \mathcal{O}(2^n)$, whereas the number of parameters of an efficient ansatz usually scales as $N_{\text{ans}} = \mathcal{O}(\text{poly}(n))$. Hence, as $n \to \infty$ more often than not the optimization problem will be under-parametrized, since $N_{\text{ans}} << N_{\text{span}}$. In smaller systems, though, where the previous inequality is not guaranteed, one might eventually use more parameters than required. This will lead to an underdetermined system with infinite solutions w.r.t a portion of its parameters, giving optimizers more paths to explore, but also more ways to diverge from previous strategies. A the simple illustrative example of this behaviour can be seen in fig. 4.1. As shown in fig. 4.2 the choice of the optimizer also plays an important role, since some optimizers are more likely to make jumps to different local minima. Stochastic optimizers for example, will most certainly introduce more discontinuities that deterministic ones.

From the above, it is apparent that in order to ensure precision and suffice to simple, scalable model architectures, it is imperative to take into account any potential hindrances posed by aforementioned discontinuities, which can obstruct training and prediction to a substantial extent. Coherently, various classical optimizations schemes used in VQLS may introduce different convergence behaviours with respect to the utilized algorithm. When dealing with convergence alone, the only optimization property that matters is the execution time, where the fastest algorithm to reach a minimum loss, can be considered best for the given problem. In our case though, continuity is of equal importance, since we seek to not only reach a solution, but to force systems with close physical parameters $\mathbf{p}^i, \mathbf{p}^j \in \mathbb{R}^d$ to produce output in the ansatz parameter space $\boldsymbol{\alpha}^i_{\text{opt}}, \boldsymbol{\alpha}^j_{\text{opt}} \in \mathbb{R}^r$ in the same neighbourhood. An optimization algorithm that favours explorability instead of exploitation has a higher chance to visit different local minima during the optimization of concurrent, similar QLSPs and thus resulting in optimal parameter discontinuities with higher probability. In the subsequent experiments, we were confined to gradient-free optimizers in order to evaluate their sufficiency with respect to the potential discontinuities.

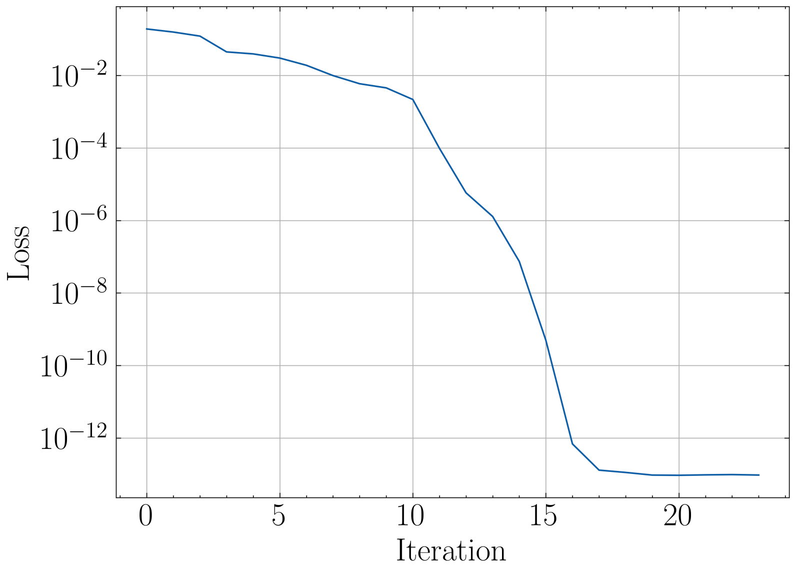


**(a)** *Optimal parameter $\alpha_0$ w.r.t. physical parameter $p$ for a $2 \times 2$ system requiring 1 parameter to span the Hilbert space.*

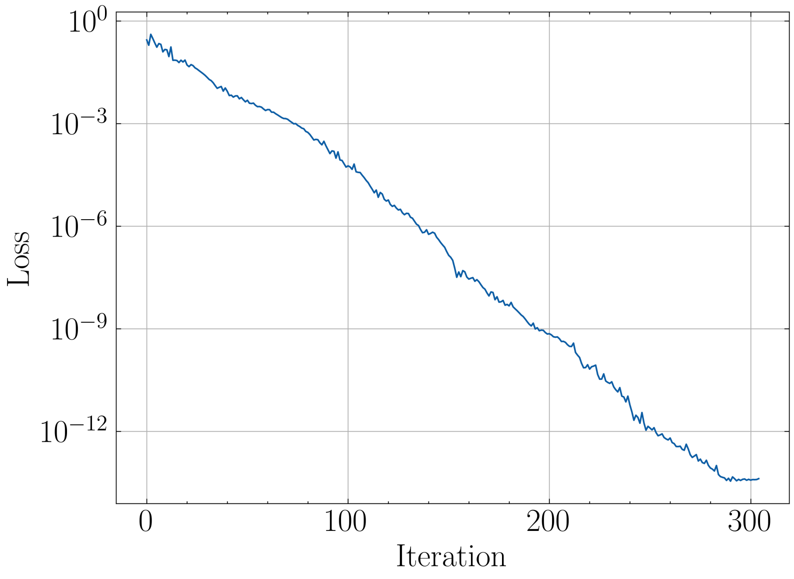


**(b)** *Optimal parameters $\alpha_0, \alpha_1$ w.r.t physical parameter $p$ for a $2 \times 2$ system.*

**Figure 4.1.** *Optimal ansatz parameters w.r.t physical parameteres for $2 \times 2$ single-qubit system using the COBYLA method. The discontinuities in fig. 4.1a are due to the periodicity of the non-convex Hamiltonian loss which allow jumps between different local minima. In fig. 4.1b which has more than required ansatz parameters, the system becomes underdetermined resulting in more discontinuities.*



**(a)** *BFGS optimizer favouring exploitation.*



**(b)** *COBYLA optimizer favouring explorability.*

**Figure 4.2.** *Example of VQLS Hamiltonian loss $C_L$ for a $2^2 \times 2^2$ system per iteration. Note that BFGS algorithm uses internal iterations which are not shown in order to approximate the Hessian matrix.*

## 4.2 Example 1: two-qubit system

The first example consists of the QLSP in eq. 4.1 for $n = 2$, which results in a parametric system of size $4\times4$, with the coefficients $c_{ij}$ chosen as trigonometric functions of the physical parameters $p_1, p_2, p_3$. Three strategies were implemented: in the first two experiments the basic VQLS scheme was used with constant, $\boldsymbol{\alpha}_0^i = \mathbf{0}$, and uniformly random, $\boldsymbol{\alpha}_0^i \sim \mathcal{U}_{[0,1]}$, proposed initial parameters respectively, while in the third experiment a naive nearby scheme was utilized, where the proposed initial parameters for each linear system were the optimal parameters of the previous one $\boldsymbol{\alpha}_0^i = \boldsymbol{\alpha}_{\text{opt}}^{i-1}$.

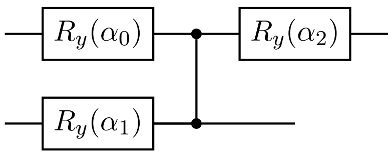


**Figure 4.3.** *Ansatz consisting of three parametric gates with angles* $\alpha_k$, $k = 0, 1, 2$ *for the* $2^2 \times 2^2$ *parametric QLSP example of 4.2.*

For the solution of the systems a BFGS optimizer was used with numerical first and second order derivatives approximation. In Fig. 4.8 one can see the joint probability distributions of the optimal parameters $\alpha_{\text{opt},k}^i$ for $k = 0, 1, 2$, $i = 1, \ldots, 1000$ for the $2^2 \times 2^2$ example of this section. In this example the dataset had no discontinuities which led to a trainable ML model.

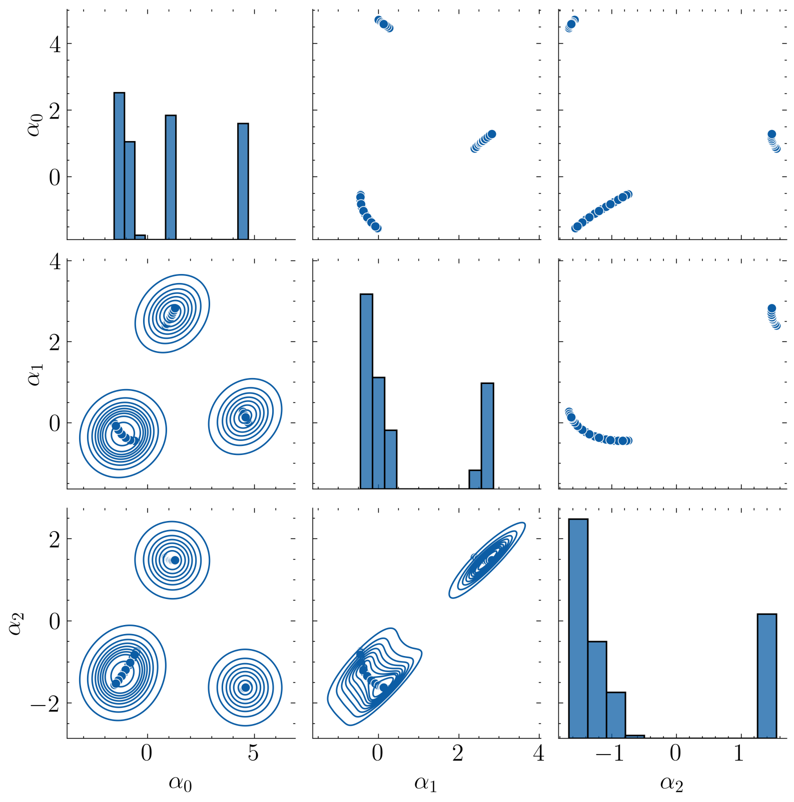


**Figure 4.4.** *Pairwise joint probability distributions of optimal parameters* $\alpha_{\text{opt},k}^i$, $k = 0, 1, 2,$ $i = 1, \ldots, 1000$ *for example 4.2.*

The latter approach takes advantage of the continuity of the LSP and requires that $\mathbf{p}^i$ are mostly sorted at each dimension in order to avoid large gaps between the sought optimal ansatz parameters. This scheme is quite efficient in the sense that it does not require any training, $N_{\text{train}} = 1$, prior to its application and can be compared in a direct manner with the base VQLS. Comparative results for the three methods are summarized in Fig. 4.3. The boxplot shows the distribution of required iterations by BFGS optimizer for solving one instance of the parametric QLSP described in 4.1.

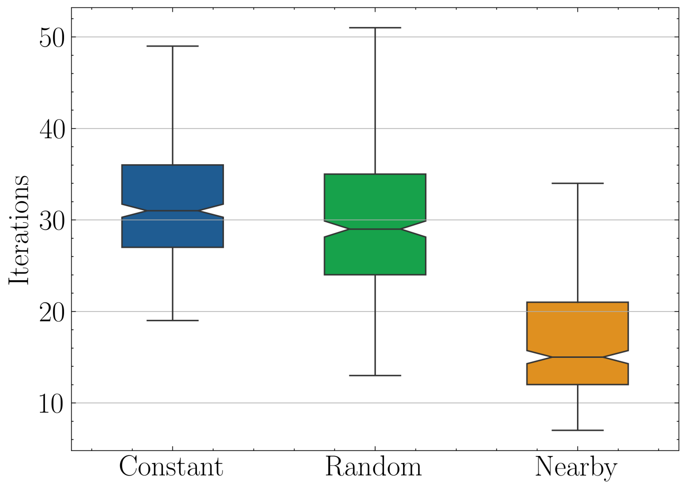


**Figure 4.5.** *Quartiles for the number of BFGS iterations required to solve an instance* $A(\mathbf{p}^i)\,|x(\boldsymbol{\alpha}^i)\rangle = |b\rangle$ *where* $A(\mathbf{p}^i) \in \mathbb{C}^{4\times 4}$. *On the left is the pure-VQLS scheme with constant initial ansatz parameters, in the middle a similiar approach but with random initialization and on the right the naive nearby approach, where the proposed ansatz parameters for each system is taken as the optimal of the previous. The speedup factor of the latter over first two schemes based on their median values is* $S \approx 1.9$.

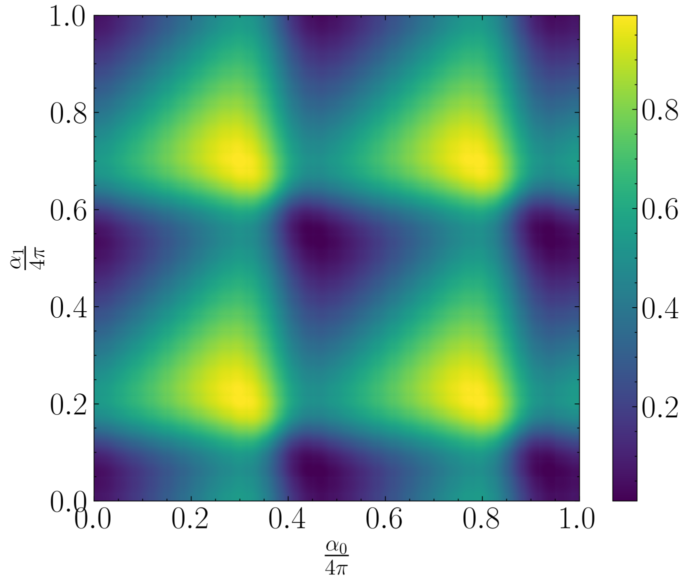


**Figure 4.6.** *Hamiltonian loss function $C_L$ w.r.t. input ansatz parameters $\alpha_0, \alpha_1$ for example 4.2. We observe an intensive non-convex behariour and periodicity resulting in different local minima which may hinder gradient-free optimization algorithms to consistently extract continuous optimal solutions $\boldsymbol{\alpha}^i_{opt}$ across a multitude of similar linear systems.*

## 4.3 Example 2: three-qubit system

For the second example, a three-qubit system was chosen, which results in a parametric system of size $8 \times 8$, with the coefficients $c_{ij}$ being polynomial functions of the physical parameters $p_i$. Three distinct strategies were compared, the first two as in example 4.2 i.e. a pure VQLS scheme with constant initial parameters and the naive nearby scheme. Additionally, as a third approach a simple MLP with 2 hidden layers was trained in the first $N_{\text{train}} = 100$ samples. In contrast to the nearby scheme, the order of solution-parameters used for training must be shuffled for the best possible exploration of paramater-space.

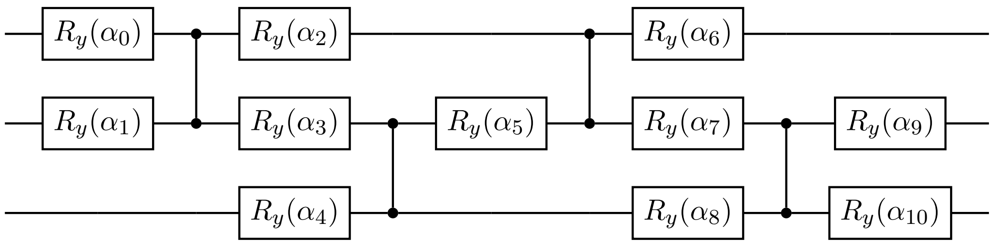


**Figure 4.7.** *Fixed-structure layered Ansatz with 11 parametric gates with angles $\alpha_k,\ k = 0, \dots, 10$ for the $2^3 \times 2^3$ parametric QLSP example of 4.2.*

The 2-layered ansatz used for this example takes as input a vector $\boldsymbol{\alpha} \in \mathbb{R}^{11}$. Each angle $a_k,\ k = 0, \dots, 10$ is used as a parameter for the $R_y$ gates which are structured as shown in Fig. 4.7. For the solution of the systems a BFGS optimizer was used with numerical first and second order derivatives approximation. In Fig. 4.8 one can see the joint probability distributions of the optimal parameters $\alpha^i_{\text{opt},k}$ for $k = 0, 1, 2$ and $i = 1, \dots, 1000$ for the

$2^3 \times 2^3$ example of this section. In this example the dataset had no discontinuities which led to a trainable ML model.

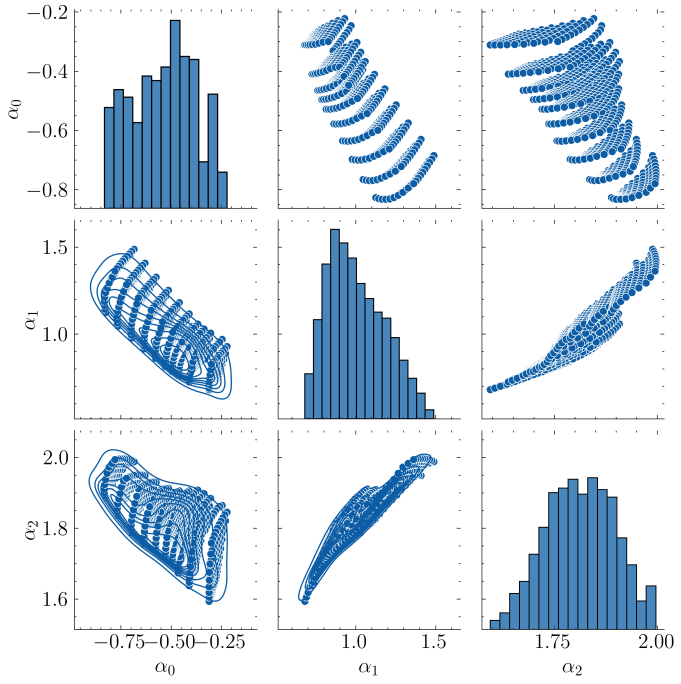


**Figure 4.8.** *Pairwise joint probability distributions of optimal parameters* $\alpha^i_{\text{opt},k}$, $k = 0, 1, 2,\ i = 1, \ldots, 1000$ *for example 4.3. In this example there are no discontinuities resulting in easier training of ML models.*

The numerical results of the three simulations are summarized in Fig. 4.9. For each of the three schemes, the graph shows the quartile distribution of the number of BFGS iterations required to reach a minimum Hamiltonian loss of at least $C_L^{\text{min}} = 10^{-12}$. The two proposed methods had almost identical performance and using the medians as reference required approximately 1/3 of the original-scheme iterations. Unlike expectations, the MLP model didn't outperform the naive Nearby method, with the mean square error, after an initial drop, usually being trapped in plateaus which were difficult to overcome without hyperparameter fine tuning, which should be avoided unless it can be automated efficiently. Since our experiments were restricted to quantum simulations, we were quite conservative in this regard. To our knowledge, unless one can use real quantum hardware and time metrics, it is not apparent how to take into account and compare the required MLP training time with respect to overall VQLS iterations. The assumption eventually made was that by using scalable models and as $N \to \infty$, which is the field quantum algorithms are supposed to excel, the model training time will be negligible with respect to the total solution time $t_{\text{train}}/t_{\text{total}} \to 0$.

With regard to low MLP performance, one can assume that factors such as discontinuities, which undermine the predictive ability of simple models played the most important role. Furthermore, the majority of the optimized parameters dataset $\boldsymbol{\alpha}^i$ cannot be used for training, as is common practice, since increasing the training data means that more solutions will be generated using the base VQLS scheme which is slower.

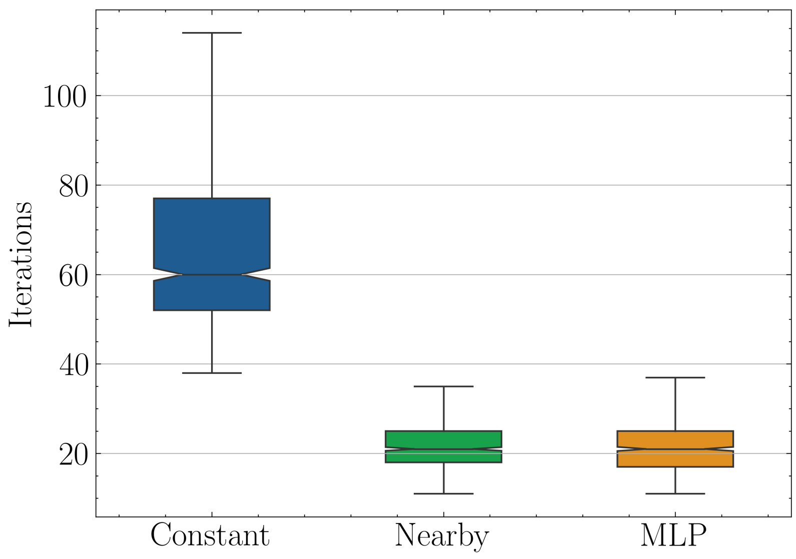


**Figure 4.9.** *Quartiles for the number of BFGS iterations required to solve QLSP instance $A(\mathbf{p}^i)\left|x(\boldsymbol{\alpha}^i)\right\rangle = |b\rangle$, where $A(\mathbf{p}^i) \in \mathbb{C}^{8\times 8}$. Like before, for the first experiment (left) the pure-VQLS scheme was with constant initial ansatz parameters, and for the second (middle) the naive nearby approach. For the last experiment (right) an MLP with 2 hidden-layers was trained with optimized ansatz parameters $\boldsymbol{\alpha}^i$ for $i = 1, \dots, N_{train} = 100$. The last two options seemingly provide the same level of acceleration with speedup factor over the original algorithm being $S \approx 2.9$ based on the median values.*

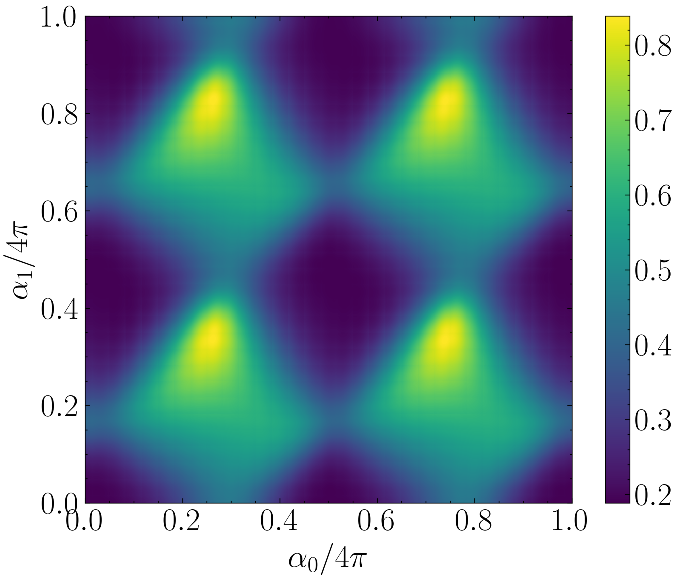


**Figure 4.10.** *Hamiltonian loss function $C_L$ w.r.t. input ansatz parameters $\alpha_0, \alpha_1$ for example 4.3. As in the first example, the Hamiltonian is non-convex resulting in different local minima, each one with similar loss values.*

## 4.4 Conclusions and future work

Quantum algorithms for Linear Systems promise to solve the QLSP in exponential speedup over classical algorithms. Successfully solving a QLSP does not include the retrieval of the information contained in the solution vector which is stored in an unobservable quantum state. The extraction of solution properties is left to the respective application, which means that if one needs the extract the whole solution vector, loses the quantum speedup. It is, thus, important to note that quantum algorithms for linear systems are supposed to be used as subroutines to other quantum algorithms which will take the solution state as input and proceed to further operations.

Variational algorithms — specifically in our context VQLS algorithm — are designed for the noisy qubits of the current NISQ era. In order to solve the QLSP, VQLS optimizes classically a parametric ansatz w.r.t a quantumly computed Hamiltonian cost function. Our proposed machine-learning scheme aims to accelerate the convergence of variational quantum algorithms when they are utilized as solvers in parametric problems by taking advantage of continuity. As a proof of concept, we showed that there is potential in that direction, since we managed to achieve a speed-up factor of up to 2.9. In this regard, the nearby shceme which combines both speed-up and simplicity seems to be preferable. We also demonstrated that when using gradient-free solvers, and especially ones which favour explorability, discontinuities in the optimal ansatz parameters emerge for various reasons, most prominently due to non-convexity of Hamiltonian loss and periodicity of ansatzes w.r.t. their paramaters. This considerably handicaps the performance of machine learning models such as MLPs, a problem that should be addressed in the future in order to take full advantage of modern ML models.

A natural next step in order to overcome those hindrances is to use exact gradient-informed optimizers, which shall favour exploitation over exploration of parameter-space. Additionally, using domain constraint optimizers will help alleviate discontinuities related to the periodicity of the Hamiltonian. In particular, the use of natural gradient methods initially proposed in [38] for classical neural networks and their quantum counterparts [39] may provide sufficient robustness to the optimation process and in this way addressing the discontinuities originating from non-convex loss functions. An extensive evaluation of transformation-preprocessing techniques may also help significantly in this regard. Without doubt, increasing the scale of the experiments will allow bigger training datasets and will enable more complex, deeper machine learning models to be trained. Finally, experiments with real quantum hardware noise must be conducted so that similar obstructive phenomena are highlighted and addressed.

# Bibliography

[1] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles, "Variational quantum linear solver," 2019.

[2] Y. Zheng, C. Song, M.-C. Chen, B. Xia, W. Liu, Q. Guo, L. Zhang, D. Xu, H. Deng, K. Huang, Y. Wu, Z. Yan, D. Zheng, L. Lu, J.-W. Pan, H. Wang, C.-Y. Lu, and X. Zhu, "Solving systems of linear equations with a superconducting quantum processor," *Phys. Rev. Lett.*, vol. 118, p. 210504, May 2017.

[3] Y. Lee, J. Joo, and S. Lee, "Hybrid quantum linear equation algorithm and its experimental test on IBM quantum experience," *Sci. Rep.*, vol. 9, p. 4778, Mar. 2019.

[4] J. Pan, Y. Cao, X. Yao, Z. Li, C. Ju, H. Chen, X. Peng, S. Kais, and J. Du, "Experimental realization of quantum algorithm for solving linear systems of equations," *Phys. Rev. A*, vol. 89, p. 022313, Feb 2014.

[5] Y. Subasi, R. D. Somma, and D. Orsucci, "Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing," *Phys. Rev. Lett.*, vol. 122, p. 060504, Feb 2019.

[6] J. Wen, X. Kong, S. Wei, B. Wang, T. Xin, and G. Long, "Experimental realization of quantum algorithms for a linear system inspired by adiabatic quantum computing," *Phys. Rev. A*, vol. 99, p. 012320, Jan 2019.

[7] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New Journal of Physics*, vol. 18, p. 023023, feb 2016.

[8] E. Anschuetz, J. Olson, A. Aspuru-Guzik, and Y. Cao, "Variational quantum factoring," in *Quantum Technology and Optimization Problems* (S. Feld and C. Linnhoff-Popien, eds.), (Cham), pp. 74–85, Springer International Publishing, 2019.

[9] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Phys. Rev. Lett.*, vol. 103, p. 150502, Oct 2009.

[10] A. Ambainis, "Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations," 2010.

[11] A. M. Childs, R. Kothari, and R. D. Somma, "Quantum algorithm for systems of linear equations with exponentially improved dependence on precision," *SIAM Journal on Computing*, vol. 46, no. 6, pp. 1920–1950, 2017.

[12] S. Chakraborty, A. Gilyén, and S. Jeffery, "The power of block-encoded matrix powers: Improved regression techniques via faster hamiltonian simulation," 2019.

[13] L. Wossnig, Z. Zhao, and A. Prakash, "Quantum linear system algorithm for dense matrices," *Phys. Rev. Lett.*, vol. 120, p. 050502, Jan 2018.

[14] X.-D. Cai, C. Weedbrook, Z.-E. Su, M.-C. Chen, M. Gu, M.-J. Zhu, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan, "Experimental quantum computing to solve systems of linear equations," *Phys. Rev. Lett.*, vol. 110, p. 230501, Jun 2013.

[15] J. Pan, Y. Cao, X. Yao, Z. Li, C. Ju, H. Chen, X. Peng, S. Kais, and J. Du, "Experimental realization of quantum algorithm for solving linear systems of equations," *Phys. Rev. A*, vol. 89, p. 022313, Feb 2014.

[16] C. Kokail, C. Maier, R. van Bijnen, T. Brydges, M. K. Joshi, P. Jurcevic, C. A. Muschik, P. Silvi, R. Blatt, C. F. Roos, and P. Zoller, "Self-verifying variational quantum simulation of lattice models," *Nature*, vol. 569, pp. 355–360, may 2019.

[17] J. Romero, J. P. Olson, and A. Aspuru-Guzik, "Quantum autoencoders for efficient compression of quantum data," *Quantum Science and Technology*, vol. 2, p. 045001, aug 2017.

[18] B. Koczor, S. Endo, T. Jones, Y. Matsuzaki, and S. C. Benjamin, "Variational-state quantum metrology," *New Journal of Physics*, vol. 22, p. 083038, aug 2020.

[19] H.-Y. Huang, K. Bharti, and P. Rebentrost, "Near-term quantum algorithms for linear systems of equations with regression loss functions," *New Journal of Physics*, vol. 23, 2019.

[20] A. Montanaro and S. Pallister, "Quantum algorithms and the finite element method," *Phys. Rev. A*, vol. 93, p. 032324, Mar 2016.

[21] J. Zhang, F. Feng, and Q. J. Zhang, "Quantum method for finite element simulation of electromagnetic problems," in *2021 IEEE MTT-S International Microwave Symposium (IMS)*, pp. 120–123, 2021.

[22] A. Mielke and T. Ricken, "Finite element analysis of a 2d cantilever on a noisy intermediate-scale quantum computer," *PAMM*, vol. 21, no. 1, p. e202100246, 2021.

[23] Y. Y. Liu, Z. Chen, C. Shu, S. C. Chew, B. C. Khoo, X. Zhao, and Y. D. Cui, "Application of a variational hybrid quantum-classical algorithm to heat conduction equation and analysis of time complexity," *Physics of Fluids*, vol. 34, p. 117121, nov 2022.

[24] D. W. Berry, "High-order quantum algorithm for solving linear differential equations," *Journal of Physics A: Mathematical and Theoretical*, vol. 47, p. 105301, feb 2014.

[25] N. Linden, A. Montanaro, and C. Shao, "Quantum vs. classical algorithms for solving the heat equation," *Communications in Mathematical Physics*, 2020.

[26] T. Xin, S. Wei, J. Cui, J. Xiao, I. n. Arrazola, L. Lamata, X. Kong, D. Lu, E. Solano, and G. Long, "Quantum algorithm for solving linear differential equations: Theory and experiment," *Phys. Rev. A*, vol. 101, p. 032307, Mar 2020.

[27] B. Zanger, C. B. Mendl, M. Schulz, and M. Schreiber, "Quantum algorithms for solving ordinary differential equations via classical integration methods," *Quantum*, vol. 5, p. 502, jul 2021.

[28] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press, 2011.

[29] N. S. Yanofsky and M. A. Mannucci, *Quantum Computing for Computer Scientists.* Cambridge University Press, 2008.

[30] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, "Quantum amplitude amplification and estimation," 2000.

[31] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, "Cost function dependent barren plateaus in shallow parametrized quantum circuits," *Nature Communications*, vol. 12, mar 2021.

[32] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," 2014.

[33] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz," *Algorithms*, vol. 12, p. 34, 2017.

[34] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, "Quantum approximate optimization algorithm for maxcut: A fermionic view," *Phys. Rev. A*, vol. 97, p. 022304, Feb 2018.

[35] G. E. Crooks, "Performance of the quantum approximate optimization algorithm on the maximum cut problem," 2018.

[36] A. Kandala, A. Mezzacapo, and K. Temme, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," *Nature*, vol. 549, p. 242–246.

[37] "Qiskit: An open-source framework for quantum computing," 2021.

[38] S.-i. Amari, "Natural Gradient Works Efficiently in Learning," *Neural Computation*, vol. 10, pp. 251–276, 02 1998.

[39] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, "Quantum natural gradient," *Quantum*, vol. 4, p. 269, may 2020.

# List of Abbreviations

| | |
|---|---|
| Q... | Quantum ... |
| C... | Classical ... |
| QC | Quantum Computing |
| NISQ | Noisy Intermediate-Scale Quantum |
| HQC | Hybrid Quantum-Classical |
| LSP | Linear System Problem |
| MCI | Monte-Carlo Integration |
| POVM | Positive Operator-Valued Measurement |
| PVM | Projection-Valued Measurement |
| VQA | Variatonal Quantum Algorithm |
| VHQCA | Variatonal Hybrid Quantum-Classical Algorithm |