ENHANCED DIGITIZED ADIABATIC QUANTUM FACTORIZATION ALGORITHM USING NULL-SPACE ENCODING

Universidad Internacional Menéndez Pelayo

ENHANCED DIGITIZED ADIABATIC QUANTUM FACTORIZATION ALGORITHM USING NULL-SPACE ENCODING

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof. dr. Jeroen J.G. Geurts,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de Faculteit der Bètawetenschappen
op [defenseDay] [DATE] om [TIME xx.xx] uur
in [de aula / het auditorium] van de universiteit,
De Boelelaan 1105

by

Felip Pellicer Benedicto

geboren te [Place, Country (if not NL) of Birth]

This dissertation was approved by

Promoter:	
Prof. dr. promoter 1	Vrije Universiteit Amsterdam, The Netherlands
Copromoter:	
Prof. dr. promoter 2	Vrije Universiteit Amsterdam, The Netherlands

Dissertation Committee

Chairman:	
Rector Magnificus, Prof. dr. Jeroen J.G. Geurts	Vrije Universiteit Amsterdam, The Netherlands
Committee:	
Prof. dr. committee member 1	Vrije Universiteit Amsterdam, The Netherlands
Prof. dr. committee member 2	Affiliation, City, Country
Prof. dr. committee member 3	Affiliation, City, Country
Prof. dr. committee member 4	Affiliation, City, Country
Prof. dr. committee member 5	Affiliation, City, Country

Acknowledgments

Without a doubt, the acknowledgments are the most widely and most eagerly read part of any thesis.

Laurens
Amsterdam, January 2021

Dedicatory of this thesis...

Author

Contents

Abstract

Your abstract.

1 Introduction

Quantum computing has emerged as the most promising paradigm for solving problems that are classically intractable. One of the most celebrated achievements is the discovery of the Shor's algorithm, which demonstrated that the integer factorization problem can be solved exponentially faster on a quantum computer than with the best-known classical algorithm. However, the full realization of this algorithm is currently out of reach due to quantum hardware limitations, known as the Noisy Intermediate-Scale Quantum (NISQ) era.

The constraints imposed by the NISQ era have motivated alternative methods to attempt to solve the same problems by using fewer quantum resources. One such approach is the Adiabatic Quantum Computation (AQC), which takes advantage of the robustness of adiabatic evolution to bring the system from a known state to a final state that encodes the solution of a problem. Another approach is the Variational Quantum Algorithm (VQA), which is a hybrid algorithm that uses quantum hardware for state evolution, and classical routines for optimization purposes.

In this work, we propose and analyze a version of the Digitized Adiabatic Quantum Factorization (DAQF) algorithm that incorporates a variation in the cost Hamiltonian. By this, we aim to provide a more resource-efficient and scalable method for integer factorization.

1.1 Circuit Model of Quantum Computation

The circuit or gate-based model of quantum computation is the most widely studied framework and the foundation of many quantum algorithms, including Shor's and Grover's algorithms. In this model, computation proceeds through the application of a sequence of unitary gates, which evolve the state of the qubits in a discrete, step-wise fashion.

Mathematically, a quantum circuit implements a unitary transformation U on the initial quantum state, typically chosen as as $|0\rangle^{\otimes n}$. This transformation is decomposed into a series of quantum gates from a universal set of gates. Measurement in the computational basis is performed at the end of the circuit to extract classical information from the quantum circuit.

Gate-based quantum computation aligns well with digital control paradigms, and most existing quantum hardware platforms, such as superconducting qubits and trapped ions, are designed to implement this model. However, the depth and width of circuits that can be reliably executed on near-term devices are limited by noise, decoherence, and imperfect gate fidelities.

FELIP: add some circuit example.

1.2 Adiabatic Quantum Computation

ALAN: Talk here about adiabatic theorem and evolutions. Then, introduce the concept/definition of adiabatic quantum computation, where we define the problem Hamiltonians and etc. To this end, we can get results by Sarandy [https://doi.org/10.1007/s11128-004-7712-7] and references therein, and some discussion done in [Rev. Mod. Phys. 90, 015002 (2018)] to make our life easier.

The adiabatic theorem of quantum mechanics provides the foundation for an alternative model of quantum computation based on continuous-time evolution. Consider a time-dependent Hamiltonian $H(t)$ with a discrete and non-degenerate spectrum. Thus we can define its instantaneous eigenstates and eigenenergies by

$$H(t)\left|n(t)\right\rangle = E_n(t)\left|n(t)\right\rangle . \tag{1.1}$$

The adiabatic theorem states that a quantum system initially prepared in an eigenstate $|n(0)\rangle$ of $H(0)$, will remain in its instantaneous eigenstate $|n(t)\rangle$ throughout the evolution, provided that the spectrum remains gapped and the change is sufficiently slow [1, 2]. This last condition is generally formulated as follows:

$$\max_{0\leq t\leq T}\left|\frac{\langle k|\dot{H}\,|n\rangle}{g_{nk}}\right| \ll \min_{0\leq t\leq T}|g_{nk}|, \tag{1.2}$$

where T is the total evolution time and g_{nk} represents the energy gap between levels n and k:

$$g_{nk}(t) \equiv E_n(t) - E_k(t) \tag{1.3}$$

Despite the adiabatic theorem works for any eigenstate, in practice it is customary to focus on ground state adiabatic passages. This is because ground states are typically more robust against decoherence and thermal excitations, as they are energetically isolated from higher-energy levels. Moreover, for many computational problems, particularly those related to optimization problems, it is possible to construct a problem Hamiltonian H_P whose ground state encodes the solution to the instance under consideration.

In the adiabatic quantum computation model, the Hamiltonian is interpolated between an initial Hamiltonian H_0, with a known and easily preparable ground state, and the problem Hamiltonian H_P, according to a schedule [2]:

$$H(t) = \big[1 - s(t)\big]H_0 + s(t)H_\mathrm{P}, \quad s(t) \in [0,1], \tag{1.4}$$

where $s(t)$ is a smooth, monotonic function such that $s(0) = 0$ and $s(T) = 1$. The problem Hamiltonian H_P is designed so that its ground state corresponds to the solution of the computational task.

(Maybe talk about AQC can be efficiently simulated in the circuit model).

FELIP: Add some figure related to AQC

1.2.1 Adiabatic Quantum Annealers

ALAN: Here we discuss about the adiabatic quantum aneealers, where the problem Hamiltonian is the Ising Hamiltonian (which is the focus of our applications). To this end, maybe we can use the review paper to get some key discussions [Rev. Mod. Phys. 90, 015002 (2018)].

An important and widely studied application of adiabatic evolution in quantum devices is adiabatic quantum annealing (AQA), where the goal is to find low-energy configurations of a classical cost function mapped onto a quantum Ising Hamiltonian. In this setting, the problem Hamiltonian H_P takes the form:

$$H_\mathrm{P} = \sum_i h_i \sigma_i^z + \sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z \tag{1.5}$$

where h_i are local fields, J_{ij} represent coupling strengths between qubits, and σ_i^z are Pauli-Z operators acting on the i-th qubit. The task is to drive the system from an initial, easily preparable ground state towards the ground state of H_P, which encodes the optimal solution to the problem at hand.

In adiabatic quantum annealers, the evolution typically starts from an initial transverse field Hamiltonian

$$H_0 = \sum_i \sigma_i^x \tag{1.6}$$

The system is then evolved adiabatically as of equation (1.4). The aim is to adiabatically steer the system from the easily preparable ground state H_0 to the ground state of H_P, which encodes the solution to the problem of interest.

FELIP: Add some figure related to Quantum Annealing

1.3 Quantum Approximate Optimization Algorithm

The Quantum Approximate Optimization Algorithm (QAOA) is a hybrid quantum-classical algorithm introduced by Farhi et al. [3] to tackle combinatorial optimization problems. Its goal is to approximate the ground state of a cost Hamiltonian whose minimum encodes the optimal solution to the problem. QAOA is particularly suited to near-term quantum devices due to its shallow circuit depth and iterative variational structure, which offloads part of the computational workload to a classical optimizer.

At its core, QAOA constructs a parametrized quantum state (ansatz) by sequentially applying two alternating types of unitaries derived from two Hamiltonians:

- The cost Hamiltonian H_C, which encodes the objective function of the optimization problem. This is typically written in the form of an Ising Hamiltonian.
- The mixing Hamiltonian H_M, which introduces transitions between computational basis states to enable exploration of the solution space. A common choice is:

$$H_M = \sum_i \sigma_i^x, \tag{1.7}$$

where σ_i^x is the Pauli-X operator acting on qubit i.

The algorithm starts with the initial state $|\psi_0\rangle = |+\rangle^{\otimes n}$, which is the ground state of H_M and a uniform superposition over all computational basis states. The QAOA ansatz with p layers is constructed as:

$$\left|\psi_p(\boldsymbol{\gamma}, \boldsymbol{\beta})\right\rangle = e^{-i\beta_p H_M} e^{-i\gamma_p H_C} \dots e^{-i\beta_1 H_M} e^{-i\gamma_1 H_C} \left|+\right\rangle^{\otimes n}, \tag{1.8}$$

where $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ are real variational parameters to be optimized.

The performance of a QAOA instance is assessed by evaluating the expected value of the cost Hamiltonian in the ansatz state:

$$F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \left\langle \psi_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) \middle| H_C \middle| \psi_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) \right\rangle . \tag{1.9}$$

This expectation value serves as the cost function for the classical optimizer. The parameters $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ are iteratively updated to minimize F_p, with quantum circuits being re-evaluated at each step until convergence.

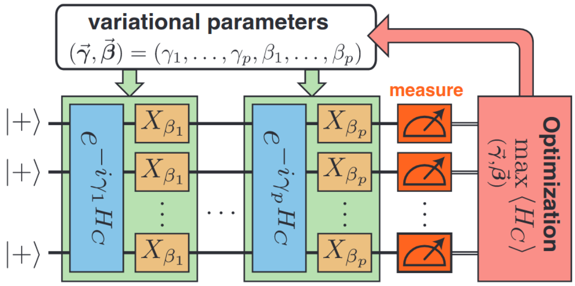

Figure 1.1: Diagram of a p-layer QAOA circuit. Starting from the initial state $|+\rangle^{\otimes n}$, the circuit alternates between applying the unitaries $e^{-i\gamma_i H_C}$ and $e^{-i\beta_i H_M}$ for $i = 1$ to p. The final state is measured to estimate the expectation value $\langle H_C \rangle$, which is passed to a classical optimizer. This process is repeated until convergence.

Source: Adapted from Zhou et al., 2019.

The QAOA stands out as one of the most compelling candidates for demonstrating quantum advantage on near-term quantum devices. From a complexity-theoretic standpoint, Farhi et al. [4] argue that sampling from the output distribution of even the lowest-depth version of QAOA (i.e., $p = 1$) could be classically intractable. Specifically, they show that there exist problem instances and choices of parameters $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ for which classical algorithms cannot feasibly reproduce the output of a quantum device running QAOA, strengthening the case for using it as a path to quantum supremacy.

Beyond theoretical hardness arguments, QAOA also exhibits practical advantages in terms of implementability and flexibility. Ho and Hsieh [5] introduce a closely related variational framework (VQCS), which shares the QAOA structure of alternating unitaries generated by simple Hamiltonians. They demonstrate that such protocols can efficiently prepare complex, non-trivial quantum states and are compatible with current quantum hardware platforms such as trapped ions and superconducting qubits. The minimal requirement of time evolution under simple, local Hamiltonians makes QAOA especially attractive for near-term experimental implementation.

Additionally, the adaptability of QAOA makes it suitable for a wide variety of problem domains, including factoring. In their work on Variational Quantum Factoring (VQF), Anschuetz et al. [6] apply a QAOA-like approach to integer factorization, showing that hybrid quantum-classical heuristics can offer a path toward solving classically hard problems on noisy intermediate-scale quantum (NISQ) devices. While challenges remain particularly related to scalability and robustness against noise the framework's modularity and reliance on classical feedback make it well-suited for real-world NISQ conditions.

2
Factorization Algorithms

In this chapter, we introduce the factorization algorithms that are most relevant to the work presented in this thesis. The integer factorization problem consists of decomposing a given semiprime number N into its prime constituents. While seemingly simple, no known classical algorithm can factor large integers efficiently, and the best classical methods –such as the general number field sieve– scale superpolynomially with input size [7]. This computational asymmetry forms the foundation of widely used cryptographic protocols, most notably RSA, which relies on the practical difficulty of factoring large semiprimes to ensure security.

The emergence of quantum algorithms has dramatically shifted the landscape of computational complexity associated with factorization. In particular, Shor's algorithm demonstrated that quantum computers can solve the problem in polynomial time, meaning a direct threat to RSA-based cryptography. However, implementing Shor's algorithm requires fault-tolerant quantum, hardware which remains out of reach in the current NISQ era.

This chapter explores two quantum approaches to the factorization problem. First, we review Shor's algorithm and its quantum Fourier transform-based structure. Then, we introduce alternative strategies based on adiabatic quantum computation, which may offer more viable paths towards factorization on near-term quantum devices.

2.1 Shor's Factorization Algorithm

ALAN: First we have to briefly introduce the Shor's algorithm. We do not need a very detailed explanation, but it would be nice to provide a didactic discussion about the algorithm and its relevance. To this end, we can use the book by Nielsen and Chuang. In this section, we make clear that it is a gate-based way to do factorization.

In computer science, an algorithm is considered to be efficient when the number of steps of the algorithm grows as a polynomial in the input size. For the problem of integer factorization, the input is a semiprime number N, and the input size is measured as the number of bits required to represent it, i.e., $\log N$. The best known classical algorithm for factoring scales superpolynomially with input size, making the problem computationally hard for large N.

$$\mathcal{O}\left(\exp\left(c(\log N)^{1/3}(\log\log N)^{2/3}\right)\right). \tag{2.1}$$

In 1994, Peter Shor introduced a quantum algorithm that factors integers in polynomial time, marking a landmark result in quantum computing. Shor's algorithm runs in time

$$\mathcal{O}\left((\log N)^{3}\right), \tag{2.2}$$

dramatically outperforming the best classical method [8]. The algorithm relies on the quantum circuit model, making it a gate-based approach to factorization and one of the strongest motivations for the development of quantum computers.

The core idea of Shor's algorithm is to reduce factorization to the problem of order finding: given a number a coprime to N, find the smallest integer, r such that

$$a^r = 1 \mod N. \tag{2.3}$$

Once the order r is known, under certain conditions, one can recover a nontrivial factor of N using elementary number-theoretic arguments [8].

The quantum part of the algorithm is used to efficiently find this order r using period-finding techniques. This is achieved by preparing a quantum superpositioin and applying the quantum Fourier transform (QFT) to extract information about the period of the function $f(x) = a^x \mod N$. The classical post-processing step then uses continued fractions to extract r and attempt to derive the prime factors of N.

Despite its theoretical significance, implementing Shor's algorithm at scale requires a large number of qubits, long coherence times, and fault-tolerant error correction, which remain beyond the reach of current quantum hardware. As such, while Shor's algorithm remains the most efficient known method for factoring in the long-term quantum regime, alternative approaches –such as adiabatic and variational algorithms– are being explored for use in the NISQ era.

2.2 Adiabatic Factorization Algorithm

The problem of integer factorization can be formulated as a constrained search over pairs of natural numbers p and q such that

$$N = p \times q. \tag{2.4}$$

In the context of adiabatic quantum computation we aim to encode this constraint into the ground state of a problem Hamiltonian, allowing the solution to emerge through adiabatic evolution.

To encode candidate solutions p and q, we adopt a binary representation of natural numbers. Any natural number N can be expressed as:

$$N = \sum_{j=0}^{n_{\text{bits}}-1} 2^j x_j, \tag{2.5}$$

where each $x_j \in \{0,1\}$ and the bit string $\mathbf{x} = x_{n_{\text{bits}}-1} \dots x_0$ represents the binary encoding of N. In our approach, we exploit the fact that the factors p and q of an odd composite number can be rewritten as:

$$\begin{cases} p = 2p' + 1 \\ q = 2q' + 1 \end{cases}. \tag{2.6}$$

It can be proved that the n_p and n_q are an upper bound on the number of qubits required to represent p' and q', respectively (FELIP: confirm that this bounds are really for p' and q'

and not for p and q):

$$\begin{cases} n_p = m\left(\lfloor\sqrt{N}\rfloor_o\right) - 1 \\ n_q = m\left(\left\lfloor\dfrac{N}{3}\right\rfloor\right) - 1 \end{cases}, \tag{2.7}$$

where $\lfloor a\rfloor$ $\left(\lfloor a\rfloor_o\right)$ denotes the largest (odd) integer not larger than a, while $m(b)$ denotes the smallest number of bits required for representing b [9]. Then, the adiabatic factorization algorithm will make use of $n = n_p + n_q$ qubits. The full quantum state

$$|\Psi\rangle = |\Psi_{p'}\rangle \otimes |\Psi_{q'}\rangle, \tag{2.8}$$

where $|\Psi_{p'}\rangle = |\psi_1\rangle \otimes \cdots \otimes |\psi_{n_p}\rangle$ and $|\Psi_{q'}\rangle = |\psi_{n_p+1}\rangle \otimes \cdots \otimes |\psi_{n_p+n_q}\rangle$.

To encode the factorization constraint into the adiabatic quantum framework, we follow the approach of Ref. [10] and define the objective function:

$$f(p,q) = \left(N - p \times q\right)^2, \tag{2.9}$$

such that its global minimum $f(p,q) = 0$ corresponds to valid factors. Translating this into a Hamiltonian acting on the computational basis, we obtain the quadratic problem Hamiltonian:

$$\hat{H}_{\mathrm{QP}} = \left[N\mathbb{1} - \left(\sum_{\ell=1}^{n_p} 2^\ell \hat{x}_\ell + \mathbb{1}\right)\left(\sum_{m=1}^{n_q} 2^m \hat{y}_m + \mathbb{1}\right)\right]^2, \tag{2.10}$$

where $\hat{x}_\ell = \dfrac{\mathbb{1} - \hat{\sigma}_\ell^z}{2}$ and $\hat{y}_m = \dfrac{\mathbb{1} - \hat{\sigma}_m^z}{2}$ are the number operators acting on the qubits encoding p' and q', respectively. The solution to the factorization problem is encoded in the ground state of $\hat{H}_{\mathrm{QP}}$.

The initial state of the system is prepared as:

$$|\psi(0)\rangle = |+\rangle^{\otimes n}, \tag{2.11}$$

where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ is the eigenstate of $\hat{\sigma}^x$, corresponding to the ground state of the initial Hamiltonian $\hat{H}_0$ defined in Eq. 1.6. Following the adiabatic theorem, if the evolution from $\hat{H}_0$ to $\hat{H}_{\mathrm{QP}}$ is slow enough, the system will remain in the instantaneous ground state, reaching the ground state of $\hat{H}_{\mathrm{QP}}$ at the end of the protocol. This final state encodes the solution to the factorization problem.

Despite its coneptual clarity, the Hamiltonian $\hat{H}_{\mathrm{QP}}$ includes high-order multi-qubit interactions, such as three- and four-body terms of the form $\hat{\sigma}_\ell^z\hat{\sigma}_m^z\hat{\sigma}_k^z$ and $\hat{\sigma}_\ell^z\hat{\sigma}_m^z\hat{\sigma}_k^z\hat{\sigma}_n^z$. These many-body interactions are difficult to implement, since they require to bring all involved qubits together and make them interact in a controled way, resulting in prone-to-error processes.

To mitigate this issue, we propose a linearized problem Hamiltonian inspired by the same factorization condition, defined as:

$$\hat{H}_{\mathrm{LP}} = N\mathbb{1} - \left(\sum_{\ell=1}^{n_p} 2^\ell \hat{x}_\ell + \mathbb{1}\right)\left(\sum_{m=1}^{n_q} 2^m \hat{y}_m + \mathbb{1}\right). \tag{2.12}$$

$\hat{H}_{\text{LP}}$ contains only two-body interaction terms, which can be efficiently simulated. However, unlike $\hat{H}_{\text{QP}}$, the desired solution is not the ground state of $\hat{H}_{\text{LP}}$, but an eigenstate with eigenvalue zero. Since the adiabatic theorem applies to any non-degenerate eigenstate, not necessarily the ground state, we hypothesize that this eigenstate can still be targeted by an appropriate adiabatic or variational algorithm.

(FELIP: maybe put the results that these research groups obtained when applying AQC for factorization)

2.2.1 Digitized Adiabatic Quantum Factorization

Although the adiabatic model is often formulated in terms of continuous time evolution, it can be simulated efficiently using gate-based quantum computation (ADD REFERENCES) through a process known as digitization.

In the digitized approach, the continuous adiabatic evolution governed by a time-dependent Hamiltonian as of Eq. 1.4 is approximated by a sequence of quantum gates through trotterization, breaking the total evolution into small time slices. Each slice is implemented as a layer in a quantum circuit, simulating the adiabatic trajectory step by step.

This method was successfully demonstrated in Ref. [10], where the authors implemented a digitized adiabatic factorization algorithm on superconducting hardware.

2.2.2 QAOA Applied to Factorization

The Quantum Approximate Optimization Algorithm can be viewed as a shortcut to adiabaticity. Rather than performing a slow, continuous evolution, QAOA uses a fixed-depth quantum circuit composed of alternating unitaries derived from the mixing and problem Hamiltonians. The parameters of these unitaries are optimized variationally to prepare a state that approximates the solution.

As shown in Ref. [11], QAOA and quantum annealing share universal structural properties, and there is compelling evidence that smooth annealing paths can be constructed from QAOA optimal parameters. This supports the interpretation of QAOA as a digitized and variationally optimized version of an adiabatic process. (FELIP: comment that this has been observed in MaxCut and QUBO problems, and provide references)

As mentioned at the beginning in the introduction of Section 2.2, the adiabatic factorization algorithm needs to deal with the difficulty of implementing three- and four-body interaction terms. In the digitized version of this problem, like QAOA, this difficulty is transformed into the necessity to implement three- and four-qubit gates. (FELIP: add figure showing one-layer QAOA circuit and the presence of 3- and 4-qubit gates).

Then, our proposal is to replace the original Hamiltonian in the QAOA layers with the linearized problem Hamiltonian defined in Eq. 2.12, while using a separate cost Hamiltonian—such as $\langle \hat{H}_{\text{QP}} \rangle$—to guide the classical optimization. This approach is motivated by a key observation: the adiabatic theorem does not constrain us to evolve within the ground state of a system, but rather within any eigenstate, provided the evolution is slow enough. We hypothesize that initializing the system in an eigenstate near the center of the spectrum of the initial Hamiltonian and evolving adiabatically under the linearized Hamiltonian may preserve this eigenstate structure, ultimately leading to the correct factorization solution, which is also located near the center of the spectrum.

In standard QAOA, the same Hamiltonian is typically used for both quantum evolution and classical cost evaluation. In contrast, our proposal explicitly decouples these two roles: we use the linearized Hamiltonian to drive the quantum dynamics (problem Hamiltonian), and a separate cost Hamiltonian—whose ground state encodes the correct factors—to steer the classical optimizer.

3

Results

This chapter we introduce our results!!! To this end, we already can state here the problem of the many-body terms presented in the previous section. Then, we can be more clear in the next section.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

3.1 QAOA-based Factorization

To exemplify our problematic, we can start by explaining how to implement the quadratic Hamiltonian. It would be interesting to show a quantum circuit to implement interactions of the form ZZ, ZZZ and $ZZZZ$. If I'm not wrong, it is already done in the Nielsen-Chuang's book. In this way, we can say that *the needed to avoid those terms if of pivotal interest to provide efficient quantum algorithms.*

4 Conclusion

The conclusions of your thesis.

4.1 This is a section title

Hello have a table.

What? A booktabs table?

4.1.1 This is a subsection

Hi again!

This is a subsubsection header

Bye this time.

Table 4.1: A nice table.

Column name	Explanation
last_modified	Kaas
version	Baas
db_schema_version	Haas

Table 4.2: A nice table.

Column name	Explanation
last_modified	Kaas
version	Baas
db_schema_version	Haas

Bibliography

URLs in this thesis have been archived on Archive.org. Their link target in digital editions refers to this timestamped version.

References

[1] M. S. Sarandy, L.-A. Wu, and D. A. Lidar. Consistency of the Adiabatic Theorem. *Quantum Information Processing*, 3(6):331–349, December 2004.

[2] Tameem Albash and Daniel A. Lidar. Adiabatic quantum computation. *Reviews of Modern Physics*, 90(1):015002, January 2018.

[3] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A Quantum Approximate Optimization Algorithm, November 2014. arXiv:1411.4028 [quant-ph].

[4] Edward Farhi and Aram W. Harrow. Quantum Supremacy through the Quantum Approximate Optimization Algorithm, October 2019. arXiv:1602.07674 [quant-ph].

[5] Wen Wei Ho and Timothy H. Hsieh. Efficient variational simulation of non-trivial quantum states. *SciPost Physics*, 6(3):029, March 2019. arXiv:1803.00026 [cond-mat].

[6] Eric R. Anschuetz, Jonathan P. Olson, Alán Aspuru-Guzik, and Yudong Cao. Variational Quantum Factoring, August 2018. arXiv:1808.08927 [quant-ph].

[7] P.L. Montgomery. A survey of modern integer factorization algorithms. *CWI Quarterly*, 7(4):337—366, December 1994.

[8] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[9] Xinhua Peng, Zeyang Liao, Nanyang Xu, Gan Qin, Xianyi Zhou, Dieter Suter, and Jiangfeng Du. A Quantum Adiabatic Algorithm for Factorization and Its Experimental Implementation. *Physical Review Letters*, 101(22):220405, November 2008. arXiv:0808.1935 [quant-ph].

[10] Narendra N. Hegade, Koushik Paul, Francisco Albarrán-Arriagada, Xi Chen, and Enrique Solano. Digitized Adiabatic Quantum Factorization. *Physical Review A*, 104(5):L050403, November 2021. arXiv:2105.09480 [quant-ph].

[11] Pablo Díez-Valle, Fernando J. Gómez-Ruiz, Diego Porras, and Juan José García-Ripoll. Universal Resources for QAOA and Quantum Annealing, June 2025. arXiv:2506.03241 [quant-ph].

GLOSSARY

FDD Feedback-Driven Development, a model of the modern code creation cycle that involves acquiring and integrating feedback from multiple sources and passing quality gates in a highly customizable way (described in Chapter 1).