# 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]

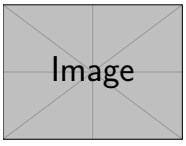
**This dissertation was approved by**

| | |
|---|---|
| *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 |

VU VRIJE UNIVERSITEIT AMSTERDAM




NWO

# 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.

## 1.2 ADIABATIC QUANTUM COMPUTATION

ALAN: Talk here about adiabatic theorem and evolutions. Then, introduce the concept/defi-

1

nition 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

$$H(t)\,|n(t)\rangle = E_n(t)\,|n(t)\rangle\,. \tag{1}$$

The adiabatic theorem states that a quantum system initially prepared in an eigenstate $|n(0)\rangle$ of $H(t)$, 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:

$$\max_{0\le t\le T}\left|\frac{\langle k|\,\dot{H}\,|n\rangle}{g_{nk}}\right| \ll \min_{0\le t\le T}|g_{nk}|, \tag{2}$$

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

$$g_{nk}(t) \equiv E_n(t) - E_k(t) \tag{3}$$

Despite the adiabatic theorem works for any eigenstate, it is customary in quantum computation 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 $H_{\mathrm{P}}$ 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 $H_{\mathrm{P}}$, according to a schedule [2]:

$$H(t) = \big[1 - s(t)\big]H_0 + s(t)H_{\mathrm{P}}, \quad s(t) \in [0,1], \tag{4}$$

where $s(t)$ is a smooth, monotonic function such that $s(0) = 0$ and $s(T) = 1$. The problem Hamiltonian $H_{\mathrm{P}}$ 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).

### 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 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:

$$H_P = \sum_i h_i \sigma_i^z + \sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z$$

where $h_i$ are local fields, $J_{ij}$ represent coupling strengths between qubits, and $\sigma_i^z$ 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

$$H_0 = -\sum_i \sigma_i^x,$$

whose ground state is a uniform superposition of all computational basis states. Then, the system is then evolved adiabatically as of equation (4).

## 1.3 QAOA

aaaaaaaaaaaaaaaaaaa

# 2

# Factorization Algorithms

This chapter we introduce the factorization algorithms of interest to our work. 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.

## 2.1 Shor's Factorization Algorithm

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.

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.

## 2.2 Adiabatic Factorization Algorithm

To end, we introduce the adiabatic factorization protocol. In this section, we make clear that it is different from the gate-based version. To finish this section, we can already mention many-body terms that appear in the adiabatic factorization method. To this section, we can use the discussions done during the meetings, and the papers you used to learn about it.

Here I include a discussion to help you to understand our goals and advances... Also, you can use it in your thesis and get inspired to explain the adiabatic theorem...

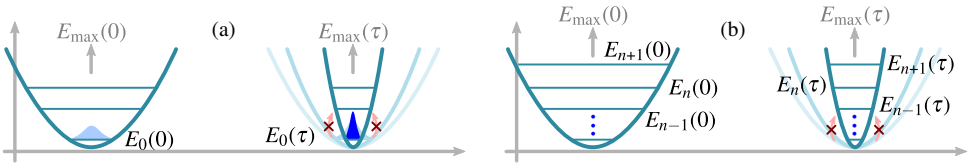



Figure 2.1: Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here.

An adiabatic algorithm works as follow. Following the standard strategy of adiabatic algorithms, the solution of the problem is encoded in the problem Hamiltonian $\hat{H}_{\rm P}$, and then we define a time-dependent adiabatic Hamiltonian $\hat{H}_{\rm ad}(t)$ such that the initial state of the system is eigenstate of $\hat{H}_{\rm ad}(t)$ at the time $t = 0$ associated to the $n$-th eigenstate $|E_n(0)\rangle$ energy level with initial energy $E_n(0)$. So, as depicted in Fig. XX, if the evolution is done through an adiabatic evolution (very slowly-varying $\hat{H}_{\rm ad}(t)$), at the end of the dynamics, say $t = \tau$, the state of the system will be $|E_n(\tau)\rangle$ which is the *update* $n$-th eigenstate of adiabatic Hamiltonian at the instant of time $t = \tau$. In particular, as shown in Fig. XX, adiabatic algorithms evolves from initial ground states because those states are protected against some kinds of decoherence, and undesired transitions can be suppressed by properly protecting couplings between the ground and first excited states. However, according to the adiabatic theorem, notice that $|E_n(0)\rangle$ does not be the fundamental state of $\hat{H}_{\rm ad}(0)$, and we could start the evolution from an arbitrary state as shown in Fig. XX. In this case, the transitions like $|E_n(t)\rangle \to |E_{n-1}(t)\rangle$ or $|E_n(t)\rangle \to |E_{n+1}(t)\rangle$ have to be avoided by properly driving the system according to the conditions to adiabaticity. The main problem is that, in those case, the ground-state protection is no longer valid because we started evolutions from states with higher energies. However, in absence of any undesired external effect, the algorithm works perfectly according adiabatic theorem and its validity conditions.

Therefore, if we know that the solution of a given problem is encoded in the $n$-th eigenstate of $\hat{H}_{\rm P}$, we just need to choose $\hat{H}_{\rm ad}(t = \tau) = \hat{H}_{\rm P}$. This can be done, for example, using the following time-dependent Hamiltonian

$$\hat{H}_{\rm ad}(t) = \left(\frac{t}{\tau}\right)\hat{H}_{\rm I} + \left(1 - \frac{t}{\tau}\right)\hat{H}_{\rm P} \tag{1}$$

where $\hat{H}_{\rm I}$ is the initial Hamiltonian we can define in the way we want to. For example, in case we start the system of $N$ qubits in a superposition of the form $|\psi(0)\rangle = |+\rangle_1 |+\rangle_2 |+\rangle_3 \cdots |+\rangle_N$, here $|+\rangle$ is the eigenstate of the Pauli matrix $\hat{\sigma}^x$, we can choose an initial Hamiltonian of the form $\hat{H}_{\rm I} = \sum_{n=1}^{N} \hat{\sigma}_n^x$, or $\hat{H}_{\rm I} = \sum_{n=1}^{N} \hat{\sigma}_n^z$. This is a free choice and it depends on the way you encode the solution of the problem in the Hamiltonian $\hat{H}_{\rm P}$.

Then, to discuss our factorization approach, we can state that we use the binary encoding. To encode a state we use the binary encoding of a natural number. To this end, we use that any natural number $n$ can be written as $n = \sum_{j=0}^{N_{\rm bits}-1} x_j \cdot 2^j$, for a bit-string values $\mathbf{x} = x_{N_{\rm bits}-1} x_{N_{\rm bits}-2} \cdots x_0$, where each $x_j$ is a dichotomic defined as $x_n \in \{0, 1\}$. Using this way to encode natural numbers, we can then introduce the adiabatic algorithm for factorization.

As explained in Ref. XX, the algorithm is used to solve the following equation

$$f(p,q) = (N - p \cdot q)^2 = 0 \tag{2}$$

with $p$ and $q$ the two number that factorizes $N$. To simplify the algorithm even more, we use the fact that $p$ and $q$ can be written as $p = 2p' + 1$ and $q = 2q' + 1$, where we can encode the information of $p'$ and $q'$ using $n_p$ and $n_q$ qubits, respectively. In this way, the total number of qubits required is $n = n_p + n_q$. For simplicity, we consider the qubits placed as

$$|\Psi\rangle = \underbrace{|\psi_1\rangle \otimes \cdots \otimes |\psi_{n_p}\rangle}_{|\Psi_{p'}\rangle} \underbrace{|\psi_{n_p+1}\rangle \otimes \cdots \otimes |\psi_{n_p+n_q}\rangle}_{|\Psi_{q'}\rangle}, \tag{3}$$

such that an arbitrary state of the system reads as

$$|\Psi\rangle = |\Psi_{p'}\rangle \otimes |\Psi_{q'}\rangle, \tag{4}$$

and the information about $p'$ and $q'$ are encoded in $|\Psi_{p'}\rangle$ and $|\Psi_{q'}\rangle$, respectively. In this way, defining the Pauli matrices such that $\hat{\sigma}^z |x\rangle = (-1)^x |x\rangle$, we can encode the solution of Eq. (2) in the ground state of the quadratic problem Hamiltonian [**?** ]

$$\hat{H}_{\rm QP} = \left[ N\mathbb{1} - \left( \sum_{\ell=1}^{n_p} 2^\ell \hat{x}_\ell + \mathbb{1} \right) \left( \sum_{m=1}^{n_q} 2^m \hat{y}_m + \mathbb{1} \right) \right]^2 \tag{5}$$

where $\hat{x}_\ell = (\mathbb{1} - \hat{\sigma}^z_\ell)/2$ and $\hat{y}_m = (\mathbb{1} - \hat{\sigma}^z_m)/2$. As for the initial Hamiltonian, in this case, the adiabatic algorithm required the initial state is $|\psi(0)\rangle = |+\rangle_1 |+\rangle_2 |+\rangle_3 \cdots |+\rangle_N$, here $|+\rangle$, because we need to give as input all possible solutions of the problem. In this case, we can choose an initial Hamiltonian of the form $\hat{H}_{\rm I} = -\sum_{n=1}^N \hat{\sigma}^x_n$, where the "$-$" sign has to be included to make sure $|\psi(0)\rangle$ is ground state of $\hat{H}_{\rm I}$. So, following the adiabatic theorem, by starting the system in the ground sate a initial Hamiltonian $\hat{H}_{\rm I}$, and if the evolution is sufficiently slow, the system will be driven through an adiabatic trajectory in which the ground state of $\hat{H}_{\rm QP}$ is the final state of the system. Therefore, the final state encodes the solution of the problem.

Maybe, again, mention the problems related to $\hat{H}_{\rm QP}$... $\hat{\sigma}^z_\ell \hat{\sigma}^z_m \hat{\sigma}^z_k$ and $\hat{\sigma}^z_\ell \hat{\sigma}^z_m \hat{\sigma}^z_k \hat{\sigma}^z_n$ terms... In order to solve this challenge, we introduce here the linearized version of $\hat{H}_{\rm QP}$ using the adiabatic theorem to support our solution.

As a first remark, notice that the Hamiltonian $\hat{H}_{\rm QP}$ is engineered in order to encode the solution of the problem in its ground state, where the power 2 is used to get an optimal solution around the value $x$ around $q \cdot p$, as depicted in Fig. XX. This is why people use the expected value of the energy of $\hat{H}_{\rm QP}$ in QAOA algorithms. But, as we know, it is not mandatory, and we can simplify the problem as follow.

The adiabatic theorem does not impose that the evolution has to start in the ground state of $\hat{H}_0$. In principle, by starting in the $n$-th eigenstate $|E_n(0)\rangle$ of $\hat{H}_0$, the system undergoes an adiabatic evolution with all population in the updated $n$-th state $|E_n(t)\rangle$, up to a global phase. Therefore, the adiabatic theorem does not forbid an adiabatic algorithm to work in case the solution is encoded in an eigenstate other than the ground state. Based on this

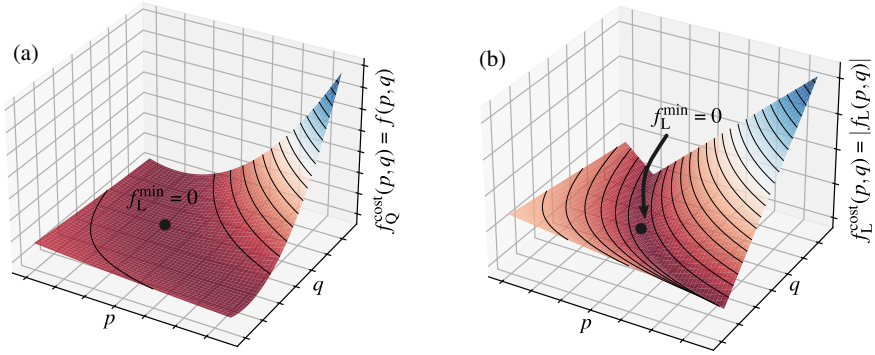



Figure 2.2: Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here.

result, we stablish the hypothesis that we can engineer a linearized problem Hamiltonian for factorization $\hat{H}_{\mathrm{LP}}$.

The Hamiltonian $\hat{H}_{\mathrm{LP}}$ is defined through the same logic used to obtain $\hat{H}_{\mathrm{QP}}$, which allows us to define

$$\hat{H}_{\mathrm{LP}} = N\mathbb{1} - \left(\sum_{\ell=1}^{n_p} 2^{\ell}\hat{x}_{\ell} + \mathbb{1}\right)\left(\sum_{m=1}^{n_q} 2^{m}\hat{y}_{m} + \mathbb{1}\right), \tag{6}$$

where the desired solution is not the ground state of $\hat{H}_{\mathrm{LP}}$, but it is the eigenstate of $\hat{H}_{\mathrm{LP}}$ with zero eigenstate. Therefore, we have to search for solutions of $|\psi_{\mathrm{out}}\rangle$ such that $\langle\psi_{\mathrm{out}}|\hat{H}_{\mathrm{LP}}|\psi_{\mathrm{out}}\rangle = 0$, which is similar to search for the zero's solution of the cost function $f_{\mathrm{L}}(p,q) = N - p\cdot q$. The problem here is that any optimization algorithms applied to this problem will provide solutions which minimizes $f_{\mathrm{L}}(p,q)$, and the minimum value of $f_{\mathrm{L}}(p,q)$ does not satisfy $f_{\mathrm{L}}^{\min} = 0$.

To bypass this problem, we exploit the freedom over the cost function $f_{\mathrm{L}}^{\mathrm{cost}}(p,q)$ defined for QAOA implementations. In fact, while the QAOA problem Hamiltonian is defined as , the cost function is defined as $f_{\mathrm{L}}^{\mathrm{cost}}(p,q) = |f_{\mathrm{L}}(p,q)|$, and therefore the output state will be driven to a solution such that minimizes $f_{\mathrm{L}}^{\mathrm{cost}}(p,q)$, that is, the solution of $|f_{\mathrm{L}}(p,q)| = 0$ that is the same solution as $f_{\mathrm{L}}(p,q) = 0$. In summary, as shown in Fig. XX, both $f_{\mathrm{L}}^{\mathrm{cost}}(p,q)$ and $f(p,q)$ have the same minimum around $p\cdot q = N$, and we can use this as input of our QAOA algorithm.

# 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.

# 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).