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 QUANTUM COMPUTATION MODEL

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.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

1.2 Adiabatic Quantum Computation

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.

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.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

1.2.1 Adiabatic Quantum Annealers

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)].

$CZ_2(\theta_n)$

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.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

1.3 QAOA

aaaaaaaaaaaaaaaaaa

2

FACTORIZATION ALGORITHMS

T his 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...
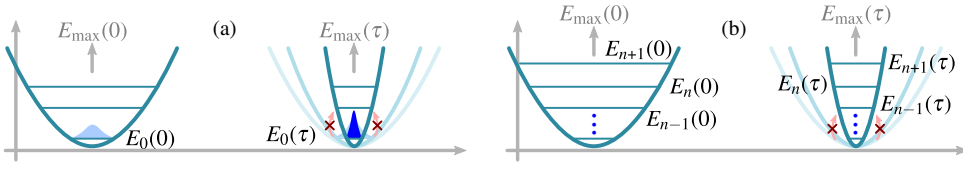
Figure 2.1: Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption 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 $\hat{H}_P$, and then we define a time-dependent adiabatic Hamiltonian $\hat{H}_{ad}(t)$ such that the initial state of the system is eigenstate of $\hat{H}_{ad}(t)$ 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 $\hat{H}_{ad}(t)$), 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 $\hat{H}_{ad}(0)$, 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 $\hat{H}_P$, we just need to choose $\hat{H}_{ad}(t = \tau) = \hat{H}_P$. This can be done, for example, using the following time-dependent Hamiltonian

$$\hat{H}_{ad}(t) = \left(\frac{t}{\tau}\right)\hat{H}_I + \left(1 - \frac{t}{\tau}\right)\hat{H}_P \tag{2.1}$$

where $\hat{H}_I$ 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 $\hat{H}_I = \sum_{n=1}^{N} \hat{\sigma}_n^x$, or $\hat{H}_I = \sum_{n=1}^{N} \hat{\sigma}_n^z$. This is a free choice and it depends on the way you encode the solution of the problem in the Hamiltonian $\hat{H}_P$.

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 $n = \sum_{j=0}^{N_{bits}-1} x_j \cdot 2^j$, for a bit-string values $\mathbf{x} = x_{N_{bits}-1} x_{N_{bits}-2} \cdots x_0$, 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

$$f(p,q) = (N - p \cdot q)^2 = 0 \tag{2.2}$$

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

$$|\Psi\rangle = \underbrace{|\psi_1\rangle \otimes \cdots \otimes |\psi_{n_p}\rangle}_{|\Psi_{p'}\rangle} \underbrace{|\psi_{n_p+1}\rangle \otimes \cdots \otimes |\psi_{n_p+n_q}\rangle}_{|\Psi_{q'}\rangle}, \tag{2.3}$$

such that an arbitrary state of the system reads as

$$|\Psi\rangle = |\Psi_{p'}\rangle \otimes |\Psi_{q'}\rangle, \tag{2.4}$$

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. (2.2) in the ground state of the quadratic problem Hamiltonian [**?**]

$$\hat{H}_{\text{QP}} = \left[N\mathbb{1} - \left(\sum_{\ell=1}^{n_p} 2^\ell \hat{x}_\ell + \mathbb{1} \right) \left(\sum_{m=1}^{n_q} 2^m \hat{y}_m + \mathbb{1} \right) \right]^2 \tag{2.5}$$

where $\hat{x}_\ell = (\mathbb{1} - \hat{\sigma}_\ell^z)/2$ and $\hat{y}_m = (\mathbb{1} - \hat{\sigma}_m^z)/2$. 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 $\hat{H}_I = -\sum_{n=1}^N \hat{\sigma}_n^x$, where the "$-$" sign has to be included to make sure $|\psi(0)\rangle$ is ground state of $\hat{H}_I$. So, following the adiabatic theorem, by starting the system in the ground sate a initial Hamiltonian $\hat{H}_I$, and if the evolution is sufficiently slow, the system will be driven through an adiabatic trajectory in which the ground state of $\hat{H}_{\text{QP}}$ is the final state of the system. Therefore, the final state encodes the solution of the problem.

Maybe, again, mention the problems related to $\hat{H}_{\text{QP}}$... $\hat{\sigma}_\ell^z \hat{\sigma}_m^z \hat{\sigma}_k^z$ and $\hat{\sigma}_\ell^z \hat{\sigma}_m^z \hat{\sigma}_k^z \hat{\sigma}_n^z$ terms... In order to solve this challenge, we introduce here the linearized version of $\hat{H}_{\text{QP}}$ using the adiabatic theorem to support our solution.

As a first remark, notice that the Hamiltonian $\hat{H}_{\text{QP}}$ 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 $\hat{H}_{\text{QP}}$ 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 result, we stablish the hypothesis that we can engineer a linearized problem Hamiltonian for factorization $\hat{H}_{\text{LP}}$.
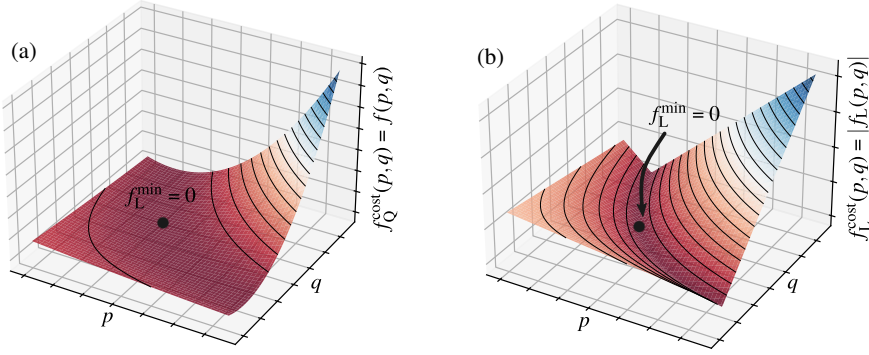
Figure 2.2: Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here. Put your caption here.

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

$$\hat{H}_{\text{LP}} = N\mathbb{1} - \left(\sum_{\ell=1}^{n_p} 2^\ell \hat{x}_\ell + \mathbb{1} \right) \left(\sum_{m=1}^{n_q} 2^m \hat{y}_m + \mathbb{1} \right), \tag{2.6}$$

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

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

3

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

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).