PRIME FACTORIZATION USING A SYSTEM OF SPINS WITH CONTROLLED COUPLING

Bachelorarbeit

zur Erlangung des akademischen Grades
Bachelor of Science (BSc)

eingereicht an der
Fakultät für Mathematik, Informatik und Physik
der
Leopold-Franzens-Universität Innsbruck

von

Valentina Zeni

Betreuer:
Univ.-Prof. Dr. Helmut Ritsch

Institut für Theoretische Physik

Innsbruck, am 27. Juli 2018

universität
innsbruck

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt durch meine eigenhändige Unterschrift, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die wörtlich oder inhaltlich den angegebenen Quellen entnommen wurden, sind als solche kenntlich gemacht.

Ich erkläre mich mit der Archivierung der vorliegenden Bachelorarbeit einverstanden.

27. Juli 2018
Datum

Valentin Fen
Unterschrift

Abstract

Integer factorization is a problem hard to solve on a classical computer and has practical relevance for RSA-based cryptosystems. This thesis discusses the factoring problem from the perspective of adiabatic quantum computation (AQC). We show how the problem of finding the prime factors of an integer can be formulated as optimization problem. In the scheme of AQC, this problem is then mapped to the physical problem of finding the ground state of a system of interacting two-state spins. We discuss an algorithm ensuring that the Hamiltonian of the system used for the adiabatic evolution only contains local terms and quadratic interactions. This approach is demonstrated for the number 21. Two different Hamiltonians of the system are derived and the adiabatic evolution is implemented for different evolution times using the Julia library QuantumOptics.jl.

Contents

Chapter 1

Introduction

Cryptography is the study of methods of enabling two or more parties to exchange messages in disguised form so that only the intended recipients can recover the contents of the conversation. A cryptographic protocol or cryptosystem is used to map the message to be sent (plaintext) to the disguised message (ciphertext) [1]. A common class of cryptosystems are the public-key cryptosystems. These are based on two keys: encryption is performed using a public key, while decryption requires a private key only known to the message recipient. To prevent third parties from eavesdropping on the contents of the conversation the encryption stage needs to be difficult to reverse even using the enciphering (public) key [2].
Many public-key cryptosystems rely on the difficulty of performing certain tasks related to number theory in order to achieve secure data transmission. Especially, the commonly used RSA cryptographic protocol (named after the inventors Rivest, Shamir, and Adleman) depends on the presumed lack of an efficient algorithm for finding the prime factors p and q of a large integer of the form $\omega = pq$ on classical computers. The public key is produced using a large biprime, while the private key is based on its prime factors. Thus, the task of inverting the encryption stage is closely related to factoring [2]. While the tasks to be performed by the intended recipient of the message (primality testing) are of polynomial or quasi-polynomial time in the size of the input, the runtime of the state-of-the-art algorithm for prime factorization - the general number field sieve - scales sub-exponentially with the input's size. This circumstance is fundamental for the usability and safety of the RSA cryptosystem [1].
However, in 1994 Peter Shor developed a quantum algorithm for integer factorization which could solve the factorization problem efficiently, i.e. in polynomial time [3]. Shor's algorithm relies on the circuit model of quantum computing, where computations are performed by application of a sequence of universal gates to a system of qubits, and is based on the reduction of factoring to period finding, which is performed using quantum Fourier transform [2]. For factoring a composite number ω having $\lfloor \log_2(\omega) \rfloor + 1 = n$ digits, this algorithm requires $\mathcal{O}\{n^3\}$ operations [3]. The perspective of breaking the RSA cryptosystem has contributed to curbing interest and research on quantum computation

[4]. Over the last years, Shor's algorithm has been demonstrated in several experiments using liquid-state NMR [5] and photonic systems [6], as well as on ion-trap quantum computers [7]. Up to now, the largest number factored using physical realizations of Shor's algorithm was 21 [8].
An alternative approach to the factoring problem is based on adiabatic quantum computation (AQC). Adiabatic quantum computation relies on the adiabatic theorem to solve optimization problems and was proposed by Farhi et al [9]. This approach was shown to be polynomially equivalent to the standard paradigm of quantum computing [10] and has been successfully applied to many different problems from the fields of physics and computer science [11, 12]. Schaller and Schützhold provided a quantum adiabatic algorithm for factoring [13]. Experimental implementations of a simplified version of this factoring scheme on NMR processors allowed to factor biprimes up to 143 [14, 15]. Furthermore, Dridi and Alghassi managed to factor all biprimes up to 200000 on the D-Wave 2X processor using computational algebraic geometry, specifically Gröbner bases, to reduce the size of the problem [16].

The topic of this thesis is a quantum adiabatic algorithm for factoring. In chapter 2 some theoretical aspects needed for the later chapters are introduced. The algorithm itself is the topic of chapter 3 and is used in chapter 4 for the factorization of the number 21.

Chapter 2

Theoretical Concepts

In this chapter we present some ideas which represent the theoretical foundation for adiabatic quantum computation and for the algorithm described in the following chapter. First, we introduce the adiabatic theorem of quantum mechanics and give an elementary proof of it. Then, we show how the timescale for adiabaticity is closely related to the properties of the Hamiltonian's spectrum. Further, we briefly present the topic of optimization. Finally, we describe the idea of adiabatic quantum computation.

2.1 Adiabatic Theorem and Landau-Zener Tunnelling

2.1.1 Dynamics of a Quantum System

In the Schrödinger picture the dynamics of a closed quantum system is described by the time-dependent Schrödinger equation

$$i\hbar\frac{\mathrm{d}}{\mathrm{d}t}\left|\psi(t)\right\rangle = \hat{H}\left|\psi(t)\right\rangle . \tag{2.1}$$

If the Hamiltonian $\hat{H}$ of the system does not depend on time, the solution to the Schrödinger equation (2.1) for the initial state $|\psi_0\rangle$ at the initial time $t = t_0$ is given by the time-dependent state

$$\left|\psi(t)\right\rangle = \hat{U}(t,t_0)\left|\psi_0\right\rangle = e^{-i\hat{H}(t-t_0)/\hbar},\left|\psi_0\right\rangle , \tag{2.2}$$

with the unitary time evolution operator $\hat{U}(t,t_0) = e^{-i\hat{H}(t-t_0)/\hbar}$ [17]. A system prepared in the eigenstate $|\phi_n\rangle$ of the Hamiltonian satisfies the eigenvalue equation (stationary Schrödinger equation)

$$\hat{H}\left|\phi_n\right\rangle = E_n\left|\phi_n\right\rangle , \tag{2.3}$$

where E_n is the energy of the eigenstate $|\phi_n\rangle$. Under time evolution, the system will remain in such eigenstate and simply acquire the global phase $e^{-iE_n(t-t_0)}$ [17].

If the Hamiltonian depends on time, the time evolution of the system does not generally take this extremely simple form. However, if the change in the Hamiltonian occurs sufficiently slowly, the adiabatic theorem comes into play and the dynamics of the system remains relatively simple.

2.1.2 The Adiabatic Theorem

The adiabatic theorem was originally stated by Born and Fock in 1928 as follows [18]:

> A physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum.

Let us consider a time-dependent Hamiltonian $\hat{H}(t)$ that governs the state of a quantum system evolving in N-dimensional Hilbert space according to the Schrödinger equation (2.1). Suppose the Hamiltonian changes smoothly from an initial form $\hat{H}_0$ at time $t = 0$ to some final form $\hat{H}_1$ at time $t = T$. Rescaling the time coordinate

$$t \to s = \frac{t}{T}, \text{ so that } s \in [0, 1], \tag{2.4}$$

we can rewrite the Hamiltonian as a function of the dimensionless parameter s, where $\hat{H}(0) = \hat{H}_0$ and $\hat{H}(1) = \hat{H}_1$ [9]. Then, for T arbitrarily large the Hamiltonian varies arbitrarily slowly as a function of time and for each time we can define the instantaneous eigenstates and eigenvalues of $\hat{H}(s)$ as

$$\hat{H}(s) \left|\phi_i(s)\right\rangle = E_i(s) \left|\phi_i(s)\right\rangle, \tag{2.5}$$

where $E_0(s) \leq E_1(s) \leq \ldots \leq E_{\mathrm{N}-1}$ for all values of $s \in [0, 1]$. Further, an initial state $|\psi(0)\rangle = |\psi_0\rangle$ will evolve in time according to the equation

$$i\hbar \frac{\mathrm{d}}{\mathrm{d}s} \left|\psi(s)\right\rangle = T\hat{H}(s) \left|\psi(s)\right\rangle, \tag{2.6}$$

resulting from (2.1) by the chain rule [19].
Suppose $|\psi_0\rangle$ is an eigenstate of $\hat{H}_0$, for simplicity the ground state $|\phi_0(0)\rangle$. The quantum adiabatic theorem guarantees that, if the Hamiltonian changes slowly enough and the gap between the two lowest energy levels, $E_1(s) - E_0(s)$, does not vanish for all values of $s \in [0, 1]$, then

$$\lim_{T \to \infty} |\left\langle \phi_0(1) \mid \psi(1) \right\rangle| = 1, \tag{2.7}$$

i.e. the evolving state system will remain arbitrarily close to the instantaneous ground state of the Hamiltonian [9]. In particular, the final state $|\psi(1)\rangle$ will be the ground state of $\hat{H}_1$.

2.1.3 Proof of the Adiabatic Theorem

At any time t the eigenstates of a time-dependent Hamiltonian $\hat{H}(t)$ satisfy the stationary Schrödinger equation (2.3). Further, these eigenstates form a complete orthogonal system, since

$$\langle \phi_m(t) \mid \phi_n(t) \rangle = \delta_{mn}. \tag{2.8}$$

A general solution to the time-dependent Schrödinger equation takes the form

$$|\psi(t)\rangle = \sum_n c_n(t) \, |\phi_n(t)\rangle \, e^{i\theta_n(t)}, \tag{2.9}$$

with $c_n(t) = \langle \phi_n(t) \mid \psi(t) \rangle$ and the dynamic phase $\theta_n(t) = -\frac{1}{\hbar} \int_0^t E_n(t')dt'$ [20]. Here $|\phi_n(t)\rangle$ and $E_n(t)$ denote the time-dependent eigenstates and eigenvalues of $\hat{H}(t)$, respectively. Plugging this expression into (2.1), we obtain

$$\begin{aligned} i\hbar \sum_n e^{i\theta_n(t)} \left(\dot{c}_n(t) \, |\phi_n(t)\rangle + c_n(t) \, |\dot{\phi}_n(t)\rangle + i c_n(t) \, |\phi_n(t)\rangle \, \dot{\theta}_n(t) \right) = \\ = \sum_n e^{i\theta_n(t)} c_n(t) \hat{H}(t) \, |\phi_n(t)\rangle \, . \end{aligned} \tag{2.10}$$

The term on the right hand side cancels out with the last summand on the left hand side, since $\dot{\theta}_n(t) = -E_n(t)/\hbar$. Acting on both sides of the resulting equation with the bra $\langle \phi_m(t)|$ and using the orthogonality of the eigenstates (2.8), we find

$$\begin{aligned} \dot{c}_m(t) &= -\sum_n c_n(t) \, \langle \phi_m(t) \mid \dot{\phi}_n(t) \rangle \, e^{i(\theta_n(t) - \theta_m(t))} \\ &= -c_m(t) \, \langle \phi_m(t) \mid \dot{\phi}_m(t) \rangle - \sum_{n \neq m} c_n(t) \frac{\langle \phi_m(t) \mid \dot{\hat{H}}(t) \mid \phi_n(t) \rangle}{E_n(t) - E_m(t)} e^{i(\theta_n(t) - \theta_m(t))}, \end{aligned} \tag{2.11}$$

where in the last step we used that $\langle \phi_m \mid \dot{\hat{H}} \mid \phi_n \rangle = (E_n - E_m) \, \langle \phi_m \mid \dot{\phi}_n \rangle$. This last relation can be easily verified by taking the time derivative of (2.3) and multiplying the resulting expression by $\langle \phi_m|$. Thus, the coefficient of the m-th eigenstate satisfy the differential equation given by (2.11). The second term on the right hand side of the equation can be neglected assuming

$$\left| \frac{\langle \phi_m(t) \mid \dot{\hat{H}}(t) \mid \phi_n(t) \rangle}{E_n(t) - E_m(t)} \right| \ll 1, \tag{2.12}$$

for a slowly changing Hamiltonian (adiabatic approximation) [21]. Then, equation (2.11) takes the form

$$\dot{c}_m(t) = -c_m(t) \, \langle \phi_m(t) \mid \dot{\phi}_m(t) \rangle \, , \tag{2.13}$$

leading to the solution for the coefficient of the m-th eigenstate

$$c_m(t) = c_m(0) \;\; e^{i\gamma_m(t)}, \tag{2.14}$$

with the geometric phase $\gamma_m(t) = -\int_0^t \langle \phi_m(t') \mid \dot{\phi}_m(t') \rangle \, dt' \in \mathbb{R}$ [20]. This result plugged into the general solution (2.9) yields

$$|\psi(t)\rangle = \sum_n c_n(0) \, |\phi(t)\rangle \, e^{i\theta_n(t)} e^{i\gamma_n(t)}. \tag{2.15}$$

Since both the dynamic and geometric phase are real, $e^{i\theta_n(t)}$ and $e^{i\gamma_n(t)}$ are only phase factors. Therefore, a system starting out in the n-th eigenstate of the Hamiltonian and evolving under the Schrödinger equation will remain in the same eigenstate [20].

2.1.4 The Landau-Zener Formula

For practical purposes, we would like to know how the required evolution time T depends on the properties of the Hamiltonian's spectrum. This can be understood considering the probability for a non-adiabatic transition in the Landau-Zener problem (avoided level crossing), the so-called Landau-Zener formula.
The simple derivation of the Landau-Zener formula that we will discuss can be found in [22]. Let us consider a two-level system with the basis states $|0\rangle$ and $|1\rangle$ and associated energies E_0 and E_1, separated by the gap $E_1 - E_0 = \hbar\omega_0 > 0$. Putting $E_0 = 0$, the Hamiltonian for this system is

$$\hat{H}_0 = \begin{pmatrix} 0 & 0 \\ 0 & \hbar\omega_0 \end{pmatrix}. \tag{2.16}$$

Considering some additional time-dependent coupling of the two states yields (e.g. an oscillating electric field in the rotating wave approximation), the Hamiltonian takes the form

$$\hat{H} = \begin{pmatrix} 0 & \hbar\Omega^\star e^{i\omega' t} \\ \hbar\Omega e^{-i\omega' t} & \hbar\omega_0 \end{pmatrix}, \tag{2.17}$$

where $\hbar\Omega$ and ω' are the strength and frequency of the time-dependent interaction, respectively. The eigenvalues of (2.17) are

$$E_+ = \frac{\hbar\omega_0}{2} + \frac{\hbar}{2}\sqrt{\omega_0^2 + 4|\Omega|^2} \tag{2.18}$$

$$E_- = \frac{\hbar\omega_0}{2} - \frac{\hbar}{2}\sqrt{\omega_0^2 + 4|\Omega|^2}, \tag{2.19}$$

with the corresponding eigenstates $|+\rangle$ (excited state) and $|-\rangle$ (ground state). Thus, the energy levels are separated by the gap $\Delta E = E_+ - E_- = \hbar\omega_0 + \hbar\sqrt{\omega_0^2 + 4|\Omega|^2}$.

For a constant detuning $\Delta = \omega - \omega_0$, the system undergoes Rabi oscillations [23]. Accordingly, the probability per unit time for the transition from the state $|0\rangle$ into the state $|1\rangle$ is given by the rate

$$\Gamma = \Omega^2 \frac{\gamma}{\Delta^2 + \gamma^2/4}, \tag{2.20}$$

where γ is the decay rate of the Rabi oscillations. Now, let us consider the situation where the detuning changes linearly with time, i.e. $\dot{\Delta} =$ constant. Then, as described in [22] we approximate the change of $\Delta(t)$ as a succession of small discrete steps each lasting δt. In each interval δt the detuning is constant and the population dynamic of the system is described by the Rabi formula (2.20). However, by changing the Hamiltonian the Rabi oscillations get out of phase and the transition amplitudes get added together with different phases. Estimating the dephasing time, i.e. the duration in which a phase difference of 2π is accumulated between consecutive intervals, as $\tau_{\mathrm{D}} \sim \sqrt{\frac{4\pi}{\dot{\Delta}}}$ and setting $\gamma = 1/\tau_{\mathrm{D}}$, we can then compute the probability $P_0(t + \delta t)$ of being in the state $|0\rangle$ after a step δt as

$$P_0(t + \delta t) = [1 - \Gamma(t)\delta t]\, P_0(t) \approx e^{-\Gamma(t)\delta t} P_0(t). \tag{2.21}$$

In this expression $P_0(t)$ is the probability of being in the state $|0\rangle$ at time t and $\Gamma(t)$ the transition rate in the segment between t and $t + \delta t$. Integrating over the duration of the entire process $T = t_f - t_i$, we obtain the probability of remaining in the state $|0\rangle$

$$\begin{aligned} P_0(T) &= \exp\left(-\int_{t_i}^{t_f} \Gamma(t) dt\right) \\ &= \exp\left\{-\frac{2\Omega^2}{\dot{\Delta}}\left[\arctan\left(\frac{\Delta(t_f)}{\gamma/2}\right) - \arctan\left(\frac{\Delta(t_i)}{\gamma/2}\right)\right]\right\}. \end{aligned} \tag{2.22}$$

Notice that instead of the transition amplitudes the transition probabilities were added together. In the limit $\Delta(t_i) \ll \gamma/2 \ll \Delta(t_f)$ equation (2.22) yields the Landau-Zener formula

$$P_0(t_f) = \exp\left(-2\pi\Omega^2/\dot{\Delta}\right), \tag{2.23}$$

depending on the velocity of change of the detuning $\dot{\Delta}$ and on the minimum gap between the energy levels which is proportional to Ω. This gives us an estimate for the probability of tunnelling non-adiabatically from the ground state to an excited state when the system encounters an avoided level crossing during time evolution [11].
This is illustrated in figure 2.1: starting from far below resonance, the system goes through the level crossing at $\Delta = 0$ and ends up far above resonance. For reference, the spectrum of the Hamiltonian is shown as a function of the separation of the energy levels ω_0 for the uncoupled case ($\Omega = 0$, dashed lines), in which the energy levels cross. In presence of coupling ($\Omega \neq 0$, solid lines), however, crossing does not occur and there is a minimum gap $g_{\min} = 2\hbar|\Omega|$.

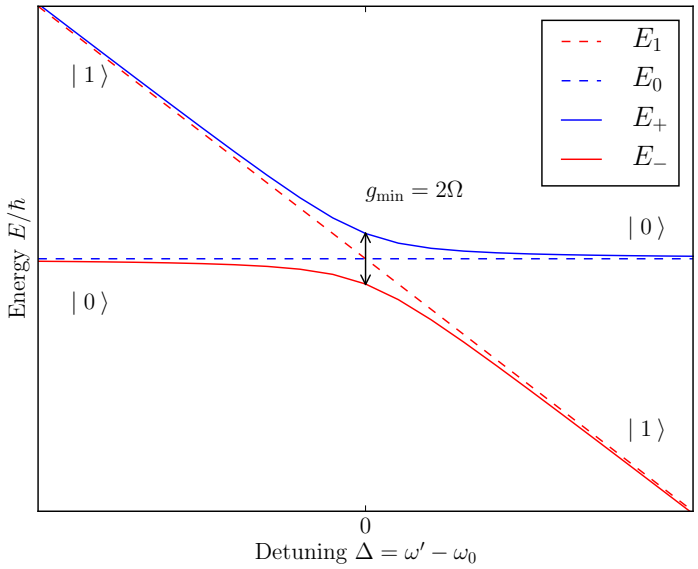


Figure 2.1: Avoided crossing in a two-level system [22]. The detuning $\Delta = \omega - \omega_0$ is swept linearly through resonance. The probability of a diabatic transition is given by the Landau-Zener formula (2.23).

Population is completely transferred from $|0\rangle$ into $|1\rangle$ if $P_0(t_f) \to 0$, corresponding to an infinitesimally slow change of Δ. Using the notation (2.4) from subsection 2.1.2, this yields the condition

$$T \gg \frac{\left|\frac{\mathrm{d}}{\mathrm{d}s}\hat{H}(s)\right|}{g_{\min}^2}, \tag{2.24}$$

where $g_{\min} = \min_{0\leq s\leq 1} E_1(s) - E_0(s)$ is the minimum gap above the ground state. If the evolution time T satisfies this condition, the adiabatic approximation is valid [9]. Thus, for a constant deformation speed of the Hamiltonian, the adiabatic runtime will approximately scale as $T = \mathcal{O}\left\{g_{\min}^{-2}\right\}$.

2.2 Optimization

Let us consider the optimization problem [24]

$$\text{minimize } f(\mathbf{x}) \quad \text{subject to } \mathbf{x} \in A, \tag{2.25}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is the real-valued function that we wish to minimize (cost function). The vector $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ contains the so-called decision variables and the

set A is a subset of $\mathbb{R}^n$ often referred to as feasible set. The task is to solve the decision problem that involves finding the vector $\mathbf{x}$ of the decision variables that minimizes the cost function over A [24]. In physics it is common to identify the cost function with the energy of system, so that the solution of the optimization problem corresponds to the ground state of said system.

2.3 Adiabatic Quantum Computation

In this section we describe the idea of quantum adiabatic computation proposed by Farhi et al. in [9] and [25]. The basic concept of adiabatic quantum computation (AQC) is to use the adiabatic theorem to solve optimization problems. The idea is to encode the solution of a minimization problem in the unknown ground state of a problem Hamiltonian $\hat{H}_\mathrm{P}$ and reach this ground state through adiabatic evolution starting from the known ground state of another Hamiltonian $\hat{H}_0$. While doing so, if the evolution time is long enough, the adiabatic theorem ensures that the evolution generates the desired state encoding the correct solution.

2.3.1 Quantum Adiabatic Evolution

The first step is to translate the optimization problem from section 2.2 in quantum-mechanical terms, i.e. find a problem Hamiltonian $\hat{H}_\mathrm{P}$ whose ground state corresponds to the solution of the problem. To find its ground state, we evolve adiabatically from the ground state of a trivial Hamiltonian $\hat{H}_0$ whose ground state is known in advance and easier to prepare. To do so, we consider the Hamiltonian $\hat{H}(t)$ smoothly interpolating between $\hat{H}(0) = \hat{H}_0$ and $\hat{H}(T) = \hat{H}_\mathrm{P}$ during the running time of the evolution T. For instance, we can interpolate linearly [9]

$$\hat{H}(t) = \left(1 - \frac{t}{T}\right)\hat{H}_0 + \frac{t}{T}\hat{H}_\mathrm{P}. \tag{2.26}$$

The system prepared in the ground state $|\psi(0)\rangle = |\psi_\mathrm{g}(0)\rangle$ of $\hat{H}_0$ will then evolve according to (2.1) into a state $|\psi(T)\rangle$ close to the solution $|\psi_\mathrm{g}(T)\rangle$. Provided the evolution is sufficiently slow, a measurement of the final quantum state state at time T will then give us with high probability the solution to our problem.

2.3.2 Physical System

Many classical computational problems (e.g. satisfiability) can be cast as the minimization of a cost function $f : \{0,1\}^n \to \mathbb{R}$, mapping n-bit strings z to real numbers. It is intuitive to identify each binary variable $z_i \in \{0,1\}$ with a spin-1/2 qubit labelled by $|z_i\rangle$ [9]. The n-qubit Hilbert space the computation is carried out on is spanned by 2^n basis

vectors $|z\rangle \equiv |z_1 z_2 \dots z_n\rangle = |z_1\rangle \otimes |z_2\rangle \otimes \dots \otimes |z_n\rangle$. Each one of the qubits lives on a two-dimensional state space with the standard computational basis $\{|0\rangle, |1\rangle\}$. We identify

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{2.27}$$

such that the state $|z_i = 0\rangle$ corresponds to the i-th spin being up and $|z_i = 1\rangle$ corresponds to the i-th spin being down [25]. We notice that the states $|z_i\rangle$ are eigenstates of the operator

$$\frac{1}{2}(\mathbb{1} - \sigma_z), \tag{2.28}$$

where $\mathbb{1}$ is the unit operator and σ_z is the Pauli z matrix $\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$. Therefore, the operator

$$\hat{z}_i \equiv \frac{1}{2}\left(\mathbb{1} - \sigma_z^{(i)}\right) \tag{2.29}$$

can be used to extract the value of the qubit it acts on:

$$\hat{z}_i |z\rangle = z_i |z\rangle. \tag{2.30}$$

2.3.3 Problem Hamiltonian and Initial Hamiltonian

A straightforward choice for the problem Hamiltonian is given according to [9] by

$$\hat{H}_{\mathrm{P}} = \sum_{z \in \{0,1\}^n} f(z) |z\rangle \langle z|. \tag{2.31}$$

This operator is diagonal in the basis of the Hilbert space and has the eigenvalues $f(z)$. Further, a common choice for the initial Hamiltonian while dealing with problems that can be formulated in terms of spin-1/2 variables is given by

$$\hat{H}_0 = -\sum_{i=1}^{n} \sigma_x^{(i)}, \tag{2.32}$$

where $\sigma_x^{(i)}$ is the Pauli x matrix $\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ acting on the i-th qubit [12]. This Hamiltonian has the ground state

$$|S\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes n}, \tag{2.33}$$

with all spins aligned in the x-direction, corresponding to an uniform superposition of all possible states.

Chapter 3

A Quantum Adiabatic Algorithm for Prime Factorization

The problem of finding the prime factors of an integer can be formulated as a classical optimization problem. This can be solved using a quantum adiabatic algorithm as the one proposed by Schaller and Schützhold [13] for the factorization of biprimes.

3.1 Factoring as Optimization Problem

The fundamental theorem of arithmetic states that every positive integer can be written as a product of positive prime numbers and that such composition is unique apart from the order of the factors [26]. Factoring means finding those prime numbers. In the following, we consider an instance of the factorization problem which is known to be particularly hard to solve, i.e. the factoring of biprimes [26]. A biprime is the product of two prime numbers. Thus, given the biprime ω, the goal is to find the prime factors a and b such that $\omega = ab$.
The factorization problem can be mapped to an optimization problem, where we minimize a cost function f defined in the most straightforward case as the square of the difference of ω and the product of the two arguments x and y

$$f\colon \mathbb{N}^2 \to \mathbb{R} \quad (x, y) \mapsto (\omega - xy)^2, \tag{3.1}$$

where x and y are positive integers [15]. This fourth degree polynomial has the required properties to be suitable as cost function for optimization, i.e. it is non-negative and vanishes at the global minima. These are at most two and are reached when $(x, y) = (a, b)$ or $(x, y) = (b, a)$ [27].
For both classical and quantum approaches to optimization it is necessary to use the binary representations of ω

$$\omega = \sum_{i=0}^{n-1} \omega_i 2^i \tag{3.2}$$

and of the factors a and b

$$a = \sum_{i=0}^{k-1} a_i 2^i, \quad b = \sum_{i=0}^{l-1} b_i 2^i, \tag{3.3}$$

where n, k, and l are the number of binary digits of ω, a, and b, respectively. The product of a k-bit number with a l-bit number has either $k + l$ or $k + l - 1$ bits [13]. Here we assume the former case and put $l = n - k$. Further, the partition $(k, n - k)$ will mostly be not known in advance. In the worst case, all possible values of k from $n/2$ to n must be tried. Verifying the answer of each try corresponds to an overhead growing linearly with n [13].
Instead of simply plugging these expressions into the cost function (3.1), it is practical to decompose long-hand multiplication into a set of factoring equations. To do so, we will use the binary multiplication table for $\omega = ab$ as described in [27] and [13]. Let us consider an example with $n = 7$ and $k = 4$. The multiplication method is shown by table 3.1. The significance of the columns increases from right to left and they are numbered starting from zero corresponding to the increasing powers of two. In table 3.1, the first two rows represent the binary digits of the multipliers a and b, followed by three rows of partial products $a_i b_j$, and two rows of carries z_{ij}, where $i < j$ indicates the carry from the i-th to the j-th column. The last row is the binary representation of the biprime ω. We can see that the number of auxiliary bits needed for the carry variables grows linearly with n.

Table 3.1: Binary multiplication table to multiply the 4-bit number a and the 3-bit number b. The significance of the columns increases from right to left. The top two rows are the binary representations of the multipliers, while the bottom row is the result ω, which is obtained summing the partial products $a_i b_j$ and carry bits z_{ij} (from the i-th to the j-th column).

Multipliers				a_3	a_2	a_1	a_0
					b_2	b_1	b_0
Binary multiplication				a_3b_0	a_2b_0	a_1b_0	a_0b_0
			a_3b_1	a_2b_1	a_1b_1	a_0b_1	
		a_3b_2	a_2b_2	a_1b_2	a_0b_2		
Carries	z_{56}	z_{45}	z_{34}	z_{23}	z_{12}		
	z_{46}	z_{35}	z_{24}				
Product	ω_6	ω_5	ω_4	ω_3	ω_2	ω_1	ω_0

This table generates a system of n factoring equations (3.4) through (3.10). Each one is obtained from the rule that the sum of the column's multiplication terms and carries from

lower significant columns is equal to the sum of the corresponding digit of the product and the carries generated in higher order columns. The resulting binary equations are

$$a_0 b_0 = \omega_0 \tag{3.4}$$
$$a_1 b_0 + a_0 b_1 = \omega_1 + 2z_{12} \tag{3.5}$$
$$a_2 b_0 + a_1 b_1 + a_0 b_2 + z_{12} = \omega_2 + 2z_{23} + 4z_{24} \tag{3.6}$$
$$a_3 b_0 + a_2 b_1 + a_1 b_2 + z_{23} = \omega_3 + 2z_{34} + 4z_{35} \tag{3.7}$$
$$a_3 b_1 + a_2 b_2 + z_{34} + z_{24} = \omega_4 + 2z_{45} + 4z_{46} \tag{3.8}$$
$$a_3 b_2 + z_{45} + z_{35} = \omega_5 + 2z_{56} \tag{3.9}$$
$$z_{56} + z_{46} = \omega_6. \tag{3.10}$$

At this point, factoring ω is equivalent to finding a solution for the binary variables $\{a_1, \ldots, a_k, b_1, \ldots, b_{n-k}, z_{ij}\}$. Then, we can introduce a cost function which is the sum of the squared factoring equations in normal form

$$f = \sum_{i=1}^{n} f_i^2, \tag{3.11}$$

where $f_i = 0$ represents the i-th factoring equation with the zero on the right hand side [13]. Therefore, each violated factoring equation translates into a positive penalty, meaning that the cost function vanishes for the correct solution, while it is strictly positive otherwise. To ensure that the variables $\{a_1, \ldots, a_k, b_1, \ldots, b_{n-k}, z_{ij}\}$ are binary it is enough to supplement (3.11) with the term

$$\sum_{i=1}^{m} w_i^2 \left(1 - w_i\right)^2, \tag{3.12}$$

where the w_i are shorthand for the involved binary variables [27]. The sum (3.12) vanishes for $w_i \in \{0, 1\}$.

3.2 Factorization in AQC

Once factorization is translated into an optimization problem, it is possible develop a quantum adiabatic algorithm to solve it. As described in chapter 2, the idea is to design a Hamiltonian whose ground state encodes the solution to the mathematical problem and evolve adiabatically into this ground state. The different ways to construct such Hamiltonian presented here are discussed in [13].

3.2.1 Physical System and Naive Hamiltonian

As stated in subsection 2.3.1, it is straightforward to identify binary variables as a spin-1/2 qubits. To start with, our goal will be to translate the simple cost function (3.1)

into a Hamiltonian. To factor a n-bit biprime ω, we consider a Hilbert space with n qubits representing the binary digits of the factors a and b defined as in (3.3). The basis states are then $|a_0\rangle \dots |a_{k-1}\rangle |b_0\rangle \dots |b_{n-k-1}\rangle$. The Hamiltonian can be obtained from the cost function replacing each of the binary variables with operators of the form (2.29) acting on the corresponding qubit. This leads to the Hamiltonian

$$\hat{H_{\mathrm{P}}}^{(1)} = \left(\omega - \hat{a}\hat{b}\right)^2, \tag{3.13}$$

where

$$\hat{a} = \sum_{i=0}^{k-1} \hat{a}_i 2^i, \quad \hat{a}_i = \frac{1}{2}\left(\mathbb{1} - \sigma_z^{(i)}\right) \tag{3.14}$$

$$\hat{b} = \sum_{i=0}^{l-1} \hat{b}_i 2^i, \quad \hat{b}_i = \frac{1}{2}\left(\mathbb{1} - \sigma_z^{(k+i)}\right). \tag{3.15}$$

The above Hamiltonian contains interactions between at most four qubits. Its couplings cover an exponential range, hence its spectral range increases exponentially with the size of the system n [13]. This makes the experimental realization harder and translates into higher computational costs.

3.2.2 Column Factoring Procedure

A better (polynomial) scaling can be achieved using the cost function (3.11) relying on the binary multiplication table. As before, the problem Hamiltonian is produced replacing all binary variables with operators (2.29) leading to

$$\hat{H}_{\mathrm{P}}^{(2)} = \sum_i E_i^2, \tag{3.16}$$

were E_i corresponds to the i-th factoring equation f_i.
This procedure requires a linear number of additional qubits for the carry variables. The advantage of $\hat{H}_{\mathrm{P}}^{(2)}$ compared to the naive Hamiltonian $\hat{H}_{\mathrm{P}}^{(1)}$ (3.13) is that the maximum penalty generated by a single equation scales as $\mathcal{O}\{(n-k)^2\}$. As a consequence, the spectral width of (3.16) scales polynomially with the number of qubits as $\mathcal{O}\{n^3\}$. However, this Hamiltonian still contains four-qubit interactions, which are difficult to realize in the experiment.

3.2.3 Cell Factoring Procedure

The same paper [13] also presents an alternative approach allowing to obtain an Hamiltonian with only quadratic interactions and coupling constants covering a finite range. The

idea is to rewrite the multiplication table 3.1 using additional auxiliary qubits in order to break each column equation into multiple smaller equations having the form

$$E_i = AB + S = 0. \tag{3.17}$$

Hereby, A and B are single bit variables and S is a sum of single bit variables. Thus, each one of these equations only contains one quadratic term AB. Actually, the penalty $P_i = E_i^2$ for violating equation (3.17) would contain three-qubit interactions. The trick is to replace it by the penalty

$$P_i' = 2\left[\frac{1}{2}\left(A + B - \frac{1}{2}\right) + S\right]^2 - \frac{1}{8}, \tag{3.18}$$

involving only quadratic interactions. It is easy to verify that both penalties vanish for the condition

$$((AB = 1) \wedge (S = -1)) \vee ((AB = 0) \wedge (S = 0)), \tag{3.19}$$

corresponding to equation (3.17) being fulfilled. The new multiplication table is listed in table 3.2.

Table 3.2: Binary multiplication table to multiply the 4-bit number a and the 3-bit number b expanded for the cell factoring algorithm. The significance of the columns increases from right to left. The top two rows are the binary representations of the multipliers, while the bottom row is the result ω. From top to bottom, each cell contains the sum of the cell directly above it S_{ij}, the partial product $a_i b_j$, and the carry variable z_{ij}. The variables satisfy $a_i b_j + S_{ij} + z_{ij} = S_{i-1,j+1} + 2z_{i+1,j}$.

Multipliers				a_3	a_2	a_1	a_0
					b_2	b_1	b_0
Top row				0	0	0	0
				a_3b_0	a_2b_0	a_1b_0	a_0b_0
				0	0	0	0
			0	S_{21}	S_{11}	S_{01}	
			a_3b_1	a_2b_1	a_1b_1	a_0b_1	
			z_{31}	z_{21}	z_{11}	0	
Bottom row		S_{32}	S_{22}	S_{12}	S_{02}		
		a_3b_2	a_2b_2	a_1b_2	a_0b_2		
		z_{32}	z_{22}	z_{12}	0		
Product	ω_6	ω_5	ω_4	ω_3	ω_2	ω_1	ω_0

Each cell contains one partial product $a_i b_j$, a sum variable S_{ij} resulting from the cell directly above it and a carry variable z_{ij} produced by the cell to its right. Therefore, each cell satisfies an equation having the form

$$a_i b_j + S_{ij} + z_{ij} = S_{i-1,j+1} + 2z_{i+1,j}, \tag{3.20}$$

where $0 \leq i \leq k-1$ and $0 \leq j \leq n-k-1$. These new factoring equations match the required form (3.17). Additional conditions are needed for the boundaries, where invalid indices occur in (3.20). These are:

$$\begin{aligned}
z_{k,j} &= S_{k-1,j+1} && \text{leftmost cell of each row} \\
S_{i-1,n-k} &= \omega_{n-k+i-1} && \text{bottom row} \\
S_{-1,j+1} &= \omega_j && \text{rightmost cell of each row} \\
S_{i,0} &= z_{i,0} = 0 && \text{top row} \\
z_{0,j} &= 0 && \text{rightmost cell of each row} \\
S_{n-k,1} &= 0 && \text{rightmost cell of the second row.}
\end{aligned} \tag{3.21}$$

The quadratic Hamiltonian constructed with the penalties (3.18) is

$$\hat{H}_{\mathrm{P}} = \sum_{i,j} \left\{ 2 \left[\frac{1}{2} \left(\hat{a}_i + \hat{b}_j - \frac{\mathbb{1}}{2} \right) + \hat{S}_{ij} + \hat{z}_{ij} - \hat{S}_{i-1,j+1} + 2\hat{z}_{i+1,j} \right]^2 - \frac{\mathbb{1}}{8} \right\}, \tag{3.22}$$

where once again each binary variable is replaced by the corresponding operator. The ground state of (3.22) is $|\psi_{\mathrm{g}}\rangle = |a_0\rangle \ldots |a_{k-1}\rangle \, |b_0\rangle \ldots |b_{n-k-1}\rangle \, |\{S_{ij}\}\rangle \, |\{z_{ij}\}\rangle$ where a and b are the correct prime factors.

Compared to the column factoring procedure from subsection 3.2.2, the cell factoring procedure requires additional ancilla variables. For all but one of the $n-k$ rows of partial products in table 3.1 k partial sum variables S_{ij} (only $k-1$ for the second row) and $k-1$ carry variables z_{ij} must be introduced. Thus, $(2k-1)(n-k-1)-1$ additional qubits are needed. Considering also the n qubits for the factors a and b, the algorithm requires at most $n-1+(2k-1)(n-k-1)$ to factorize a n-bit biprime. This number can be further reduced to $2k(n-k-1)-3$ setting the first and last bit of the prime factors to 1, as shown in [13]. Overall, this corresponds to a quadratic scaling with respect to the number of bits of the number to be factored. Also, each factoring equations contains at most six binary variables. Therefore, the number of quadratic interactions required scales quadratically in n, too. Further, the coupling strength covers a finite range from 1 to 8 and the spectral width scales only quadratically with n, opposed to the previously discussed cases [13].

Chapter 4

Example: Factoring 21

In this chapter we use the procedures discussed in chapter 3 to factorize the number 21.

4.1 Construction of the Problem Hamiltonian

The binary representation of 21 is 10101, so $n = 5$. Thus, to represent its prime factors a total of 5 or 6 bits is required. Here we assume that the factors a and b are odd and have $k = 3$ and $l = 2$ bits, respectively.
First, we use the simple approach described in subsection 3.2.1. We can directly define the Hamiltonian as in equation (3.13)

$$\hat{H}_{\mathrm{P}}^{(1)} = \left[21\mathbb{1} - \left(2^2\hat{a}_2 + 2\hat{a}_2 + \mathbb{1}\right)\left(2\hat{b}_1 + \mathbb{1}\right)\right]^2, \tag{4.1}$$

where the operators $\hat{a}_2, \hat{a}_1, \hat{b}_1$ are obtained according to equation (2.29). Assuming the state of our system is given by $|a_2 a_1 b_1\rangle$, we can compute traces of $\hat{H}_{\mathrm{P}}^{(1)}$, $\sigma_z^{(i)}\hat{H}_{\mathrm{P}}^{(1)}$, and $\sigma_z^{(i)}\sigma_z^{(j)}\hat{H}_{\mathrm{P}}^{(1)}$ to cast this Hamiltonian into the form

$$\begin{aligned}\hat{H}_{\mathrm{P}}^{(1)} = 2101\mathbb{1} + 88\sigma_z^{(1)} + 44\sigma_z^{(2)} + 84\sigma_z^{(3)} + 20\sigma_z^{(1)}\sigma_z^{(2)} - 20\sigma_z^{(1)}\sigma_z^{(3)} - \\ - 10\sigma_z^{(1)}\sigma_z^{(3)} - 16\sigma_z^{(1)}\sigma_z^{(2)}\sigma_z^{(3)},\end{aligned} \tag{4.2}$$

where we can recognise that it contains one three-qubit interaction.
Further, we construct the quadratic Hamiltonian using the cell factoring procedure from subsection 3.2.3. The binary multiplication table associated with our problem is listed in table 4.1.

Table 4.1: Binary multiplication table for $ab = 21$, where the factors a and b are both odd and have $k = 3$ and $l = 2$ bits, respectively.

Multipliers			a_2	a_1	1
				b_1	1
Binary multiplication			a_2	a_1	1
		a_2b_1	a_1b_1	b_1	
Carries	z_{34}	z_{23}	z_{12}		
Product	1	0	1	0	1

In total, six binary variables appear in table 4.1. Three of them correspond to the unknown binary digits of the prime factors a and b, while the remaining three are required for the carry variables. Notice that only one row of carry variables is needed, since only up to three binary variables are added together. From table 4.1 we obtain the factoring equations:

$$\begin{aligned} a_1 + b_1 - 2z_{12} &= 0 \\ a_2 + a_1b_1 + z_{12} - 2z_{23} - 1 &= 0 \\ a_2b_1 + z_{23} - 2z_{34} &= 0 \\ z_{34} - 1 &= 0. \end{aligned} \tag{4.3}$$

We can reduce the number of qubits needed by solving the last factoring equation for z_{34} and setting $z_{34} = 1$:

$$\begin{aligned} a_1 + b_1 - 2z_{12} &= 0 \\ a_2 + a_1b_1 + z_{12} - 2z_{23} - 1 &= 0 \\ a_2b_1 + z_{23} - 2 &= 0. \end{aligned} \tag{4.4}$$

Only five binary variables are left and the state of our system will be described by $|a_2a_1b_1z_{12}z_{23}\rangle$. Each of the remaining equations (4.4) has at most one two-qubit interaction. Therefore, the Hamiltonian obtained by squaring them and adding them together (equation (3.16)) would contain three-qubit interactions. We get rid of them using the procedure described in subsection 3.2.3. In this case, the factoring equations (4.4) already are in the required form (3.17) to write the Hamiltonian according to equation (3.22), therefore the multiplication table does not need to be rewritten. Hence, we can directly define the penalties according to equation (3.18). This leads to the total Hamiltonian

$$\begin{aligned} \hat{H}_{\mathrm{P}}^{(2)} = \left\{2\left[-\frac{\mathbb{1}}{4} + \hat{a}_1 + \hat{b}_1 - 2\hat{z}_{12}\right]^2 - \frac{\mathbb{1}}{8}\right\} + \left\{2\left[\frac{1}{2}\left(\hat{a}_1 + \hat{b}_1 - \frac{\mathbb{1}}{2}\right) + \hat{a}_2 + \hat{z}_{12} - 2\hat{z}_{23} - \mathbb{1}\right]^2 - \right. \\ \left. - \frac{\mathbb{1}}{8}\right\} + \left\{2\left[\frac{1}{2}\left(\hat{a}_2 + \hat{b}_1 - \frac{\mathbb{1}}{2}\right) + \hat{z}_{23} - 2\mathbb{1}\right]^2 - \frac{\mathbb{1}}{8}\right\}. \end{aligned} \tag{4.5}$$

Alternatively, to reduce the complexity of the computation we can assume a fixed length for the prime factors, i.e. set also the variables a_2 and b_1 to one. The corresponding multiplication table 4.2 is listed below.

Table 4.2: Binary multiplication table for $ab = 21$, where the factors a and b are both odd and have $k = 3$ and $l = 2$ bits, respectively. Here also the most significant bit of each factor is set to one.

Multipliers			1	a_1	1
				1	1
Binary multiplication			1	a_1	1
		1	a_1	1	
Carries	z_{34}	z_{23}	z_{12}		
Product	1	0	1	0	1

The associated factoring equations simplify to

$$\begin{aligned} a_1 + 1 - 2z_{12} &= 0 \\ a_1 + z_{12} - 2z_{23} &= 0 \\ 1 + z_{23} - 2z_{34} &= 0 \\ z_{34} - 1 &= 0. \end{aligned} \tag{4.6}$$

Therefore, only four qubits instead of six are required for the computation and the dimension of the Hilbert space is reduced significantly. We define the new basis states as $|a_1 z_{12} z_{23} z_{34}\rangle$ and the problem Hamiltonian is obtained using equation (3.22).
The following sections describe the implementation and the results of the algorithm with the naive problem Hamiltonian (4.1) and with the quadratic problem Hamiltonian obtained from the multiplication table (4.5).

4.2 Implementation in QuantumOptics.jl

For the implementation of the algorithm, we use QuantumOptics.jl, an open source numerical framework written in the Julia language providing many useful functions that make it easy to simulate quantum systems. It allows to define the basis of the desired Hilbert space and to create states and operators living in it. Further, these objects can be used to perform different types of time evolution [28].
First, we need to load the library into the current workspace and to define a suitable basis for the system under consideration. Being this a spin-1/2 system, we use the function `SpinBasis(n)` to create the basis for one spin variable with spin number `n` $= 1/2$ and

combine three of such bases with the tensor product `tensor` to get the basis of the Hilbert space for the computation using the naive problem Hamiltonian (4.1).

```
# load library
using QuantumOptics
# basis for one spin-1/2
onespin = SpinBasis(1//2);
# composite basis
allspin = tensor(onespin, onespin, onespin);
```

The next step is to create the problem Hamiltonian. To do so, we first define some useful operators like the Pauli matrices and the identity operator. The corresponding functions are `sigmax(b)`, `sigmay(b)`, `sigmaz(b)`, and `identityoperator(b)` and require the basis `b` of the system the operator is acting on as argument.

```
# Pauli matrices acting on one qubit
sx = sigmaz(onespin);
sy = sigmax(onespin);
sz = sigmay(onespin);
# identity operator
id = identityoperator(allspin);
```

Further, we need operators of the form (2.29) for each of the `N` $= 3$ qubits. This can be achieved by first defining the operator acting on a single qubit and then using the function `embed(b, indices, operators)` to compute a tensor product of operators specified by the vector `operators`, acting on the subsystem indicated by the integer vector `indices`. Missing indices are filled up with identity operators.

```
# binary operator acting on one qubit
opz = 1./2*(identityoperator(onespin)-sz);
# fill with identity for the other qubits
Sz = [id, id, id]
# order: a2 a1 b1 (z12 z23)
for i=1:N
    Sz[i] = embed(allspin, i, opz)
end
```

In order to perform the adiabatic evolution, we define a total Hamiltonian `H` according to (2.26). The problem Hamiltonian `Hp` is given by (4.1) and we choose (2.32) as initial Hamiltonian `H0` for the interpolation. The scaling factor 10 allows us to obtain a larger minimum gap between the ground state and the first excited state. Also, we can directly compute the energy eigenvalues using the function `eigenstates(op)`.

```
# problem Hamiltonian
Hp = (21*id-(2*Sz[3]+id)*(4*Sz[1]+2*Sz[2]+id))^2
# initial Hamiltonian
H0 = -10*(embed(allspin, 1, sx)+embed(allspin, 2, sx)+embed(allspin, 3, sx));
# linear interpolation with s = t/T
s = linspace(0, 1, 1001);
Eall = zeros(length(s), 2^N);
# spectrum of total Hamiltonian H = (1-s)*H0+s*Hp
```

```
for i = 1:length(s)
    H = (1-s[i])*H0+s[i]*Hp;
    Eall[i,:] = eigenstates(DenseOperator(H))[1]
end
```

We can now perform the time evolution for different total evolution times `T` using the function `timeevolution.schroedinger_dynamic(tspan, psi0, f)` to integrate the time-dependent Schrödinger equation. This function requires a vector `tspan` with the points for which output should be displayed, an initial state vector `psi0`, and a function `f` returning the time-dependent Hamiltonian. As initial state we use the ground state of the initial Hamiltonian that can also be obtained with the function `eigenstates(op)`, where as `op` we use `H0`.

```
# total evolution time
T = 1
# initial state, i.e. ground state of H0
psi0 = eigenstates(DenseOperator(H0))[2][1]
# time-dependent Hamiltonian as function
function Ht(t, psi)
    return (1-t/T)*H0+t/T*Hp
end
# time evolution
tout1, psi1 = timeevolution.schroedinger_dynamic(s*T, psi0, Ht);
```

The overlap with the correct solution can be obtained as the absolute square of the scalar product with the ground state of the problem Hamiltonian.

```
# correct solution
psiL = eigenstates(DenseOperator(Hp))[2][1]
# overlap with psiL
P_sol1 = [abs.((dagger(psi1[i])*psiL)).^2 for i = 1:length(s)]
```

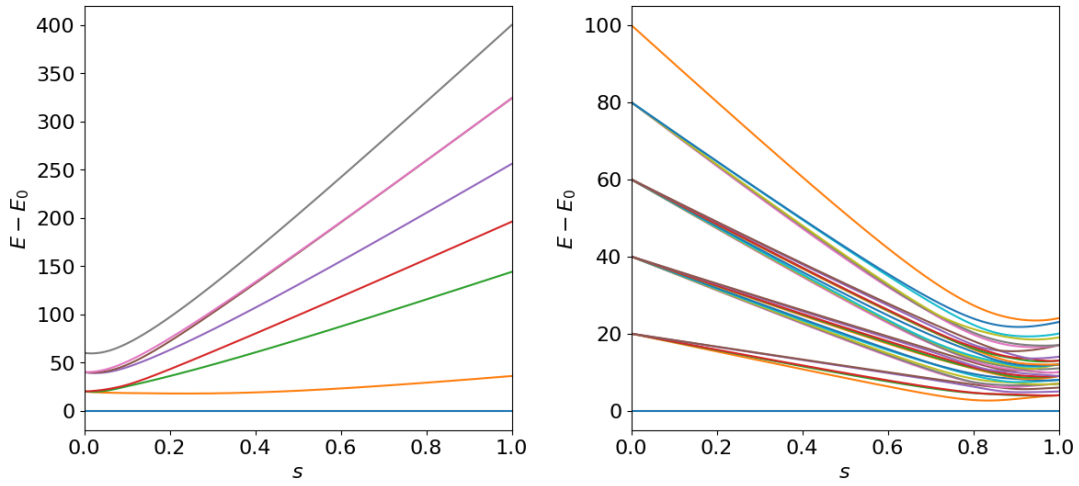
Finally, the value of the qubits can be extracted as the expectation value of the associated operator (2.29) with the function `expect(op, psi)` computing the expectation value of the operator `op` for the state `psi`.

```
# value of the qubits
display([expect(Sz[i], psi1[end]) for i = 1:N])
```

Using the quadratic Hamiltonian (4.5) instead of the naive Hamiltonian and a system of five instead of three spins, the same computation can be carried out for the cell factoring algorithm.

4.3 Results

The spectrum of the both Hamiltonians is shown in figure 4.1 in dependency of the time parameter s. The energy of the ground state was subtracted from the energy eigenvalues.

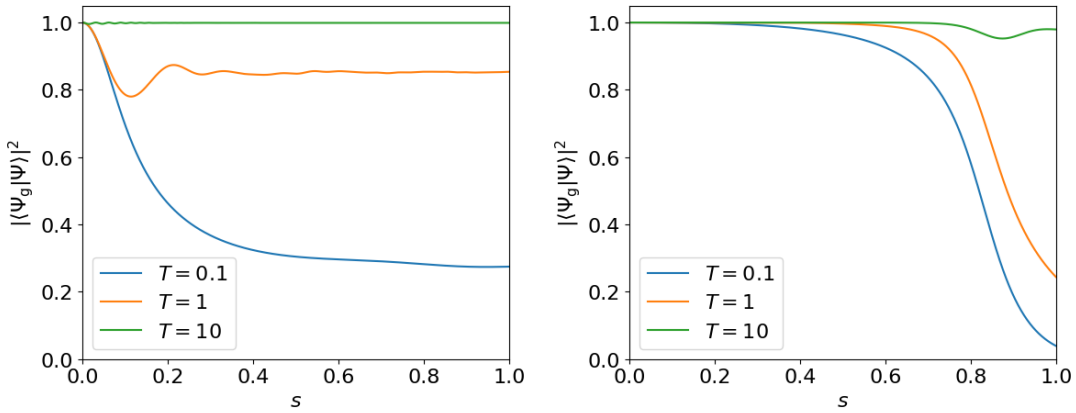


(a) naive Hamiltonian

(b) quadratic Hamiltonian

Figure 4.1: Spectrum of the total Hamiltonian for the factorization of 21. The separation of the energy eigenvalues from the lowest energy level E_0 is plotted as a function of the time parameter s for (a) the naive problem Hamiltonian $\hat{H}_{\mathrm{P}}^{(1)}$ (4.1) and (b) the quadratic problem Hamiltonian $\hat{H}_{\mathrm{P}}^{(2)}$ (4.5) obtained from the multiplication table.

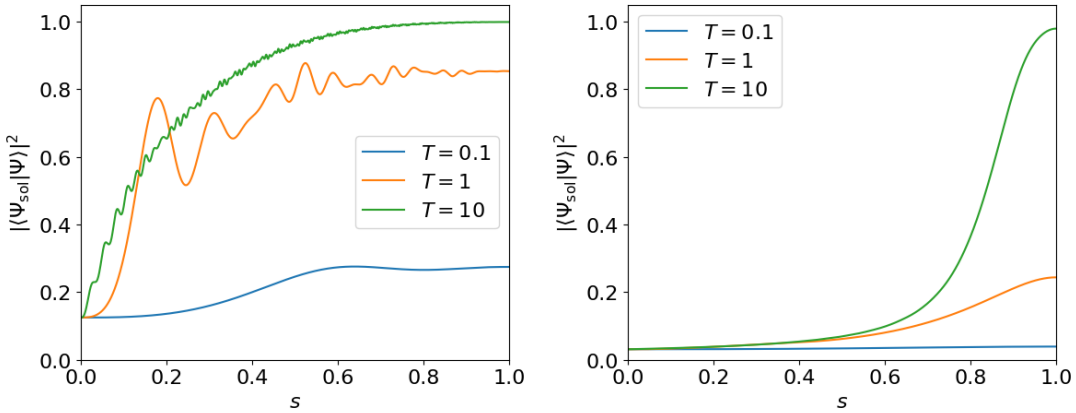
In both cases, as needed for the adiabatic evolution the ground state is non-degenerate for all $s \in [0, 1]$ and there is a gap to the excited states. The minimum gap for $\hat{H}_{\mathrm{P}}^{(1)}$ (4.1) is $g_{\mathrm{min}}^{(1)} = 17.86$ and is located at $s = 0.235$, while for $\hat{H}_{\mathrm{P}}^{(2)}$ (4.5) we find $g_{\mathrm{min}}^{(2)} = 2.65$ at $s = 0.835$. These values allow us an estimate of the required evolution time. Since $\left(g_{\mathrm{min}}^{(1)}\right)^{-2} = 0.003$ and $\left(g_{\mathrm{min}}^{(2)}\right)^{-2} = 0.143$, we carry out the time evolution for the total evolution times $T = 0.1$, 1, and 10.
Figure 4.2 shows the overlap with instantaneous eigenstate $|\psi_{\mathrm{g}}(s)\rangle$ of the Hamiltonian as an estimate for the adiabaticity of the process. We notice that for the longest evolution time - corresponding to a slower interpolation between the initial Hamiltonian and the problem Hamiltonian - the overlap is almost perfect for the whole duration: the evolution is adiabatic and the system remains close to the ground state of the Hamiltonian. In contrast, for shorter evolution times the overlap decreases quickly approaching the minimum gap, hence there is a substantial probability of finding the system in one of the excited states.



(a) naive Hamiltonian (b) quadratic Hamiltonian

Figure 4.2: Overlap with the instantaneous ground state of the total Hamiltonian $|\psi_g\rangle$ as a function of the parameter s for (a) the naive problem Hamiltonian $\hat{H}_P^{(1)}$ (4.1) and (b) the quadratic problem Hamiltonian $\hat{H}_P^{(2)}$ (4.5) obtained from the multiplication table.

Further, figure 4.3 depicts the overlap with the ground state $|\psi_{sol}\rangle$ of the problem Hamiltonian encoding the correct solution of the factoring problem. In agreement with the previous observations, the system approaches the correct solution for the slower interpolation, while for the shorter evolution times the probability for the system to be in the desired state is much smaller.



(a) naive Hamiltonian (b) quadratic Hamiltonian

Figure 4.3: Overlap with the ground state of the problem Hamiltonian $|\psi_{sol}\rangle$ as a function of the parameter s for (a) the naive problem Hamiltonian $\hat{H}_P^{(1)}$ (4.1) and (b) the quadratic problem Hamiltonian $\hat{H}_P^{(2)}$ (4.5) obtained from the multiplication table.

Finally, we obtain the value of the qubits as the expectation value of the associated operator (2.29) at the end of the evolution, e.g. for the binary variable a_2 we compute $\langle \hat{a}_2 \rangle = \langle \psi(1) \mid \frac{\mathbb{1}-\sigma_z^{(1)}}{2} \mid \psi(1) \rangle$. The results along with the correct values are listed in tables 4.3 and 4.4.

Table 4.3: Value of the qubits at the end of the time evolution with the naive problem Hamiltonian $\hat{H}_{\mathrm{P}}^{(1)}$, approximated to the second decimal place.

variable	$T = 0.1$	$T = 1$	$T = 10$
$a_2 = 1$	0.65	0.93	1.00
$a_1 = 1$	0.66	0.99	1.00
$b_1 = 1$	0.65	0.92	1.00

Table 4.4: Value of the qubits at the end of the time evolution with the quadratic problem Hamiltonian $\hat{H}_{\mathrm{P}}^{(2)}$, approximated to the second decimal place.

variable	$T = 0.1$	$T = 1$	$T = 10$
$a_2 = 1$	0.52	0.79	0.99
$a_1 = 1$	0.51	0.61	0.99
$b_1 = 1$	0.52	0.77	1.00
$z_{12} = 1$	0.50	0.61	0.99
$z_{23} = 1$	0.50	0.58	0.99

Chapter 5

Conclusions

The problem of finding the prime factors of an integer is hard to solve on a classical computer. The supposed lack of an efficient factoring algorithm is the basis for secure data transmission via RSA-based cryptographic protocols [1]. However, Shor's quantum algorithm for integer factorization is able to solve the problem in polynomial time [3]. So far, only integers up to 21 could be factored with this approach due to technical limitations.
In this thesis we showed that an alternative is offered by adiabatic quantum computation (AQC), consisting in solving optimization problems through adiabatic evolution of a suitable quantum system. As discussed in chapter 2, the core idea of AQC is to encode the solution of the problem in the ground state of a Hamiltonian and let the system evolve adiabatically into it [9]. Chapter 3 provided an overview of the quantum adiabatic factoring algorithm developed by Schaller and Schützhold [13]. Finally, this approach was demonstrated for a simple example in chapter 4.

Bibliography

[1] Neal Koblitz. *A course in number theory and cryptography*. Number 114 in Graduate texts in mathematics. Springer-Verlag, New York, 2nd ed edition, 1994.

[2] Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, Cambridge ; New York, 10th anniversary ed edition, 2010.

[3] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, 26(5):1484–1509, October 1997. arXiv: quant-ph/9508027.

[4] Aleksej Ju Kitaev, Aleksandr Ch Šen, and Michail N. Vjalyj. *Classical and quantum computation*. Number 47 in Graduate studies in mathematics. American Mathematical Society, Providence, RI, 2002. OCLC: 611603364.

[5] Lieven M. K. Vandersypen, Matthias Steffen, Gregory Breyta, Costantino S. Yannoni, Mark H. Sherwood, and Isaac L. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414(6866):883–887, December 2001. arXiv: quant-ph/0112176.

[6] B. P. Lanyon, T. J. Weinhold, N. K. Langford, M. Barbieri, D. F. V. James, A. Gilchrist, and A. G. White. Experimental Demonstration of a Compiled Version of Shor's Algorithm with Quantum Entanglement. *Physical Review Letters*, 99(25):250505, December 2007.

[7] Thomas Monz, Daniel Nigg, Esteban A. Martinez, Matthias F. Brandl, Philipp Schindler, Richard Rines, Shannon X. Wang, Isaac L. Chuang, and Rainer Blatt. Realization of a scalable Shor algorithm. *Science*, 351(6277):1068–1070, March 2016. arXiv: 1507.08852.

[8] Enrique Martín-López, Anthony Laing, Thomas Lawson, Roberto Alvarez, Xiao-Qi Zhou, and Jeremy L. O'Brien. Experimental realization of Shor's quantum factoring algorithm using qubit recycling. *Nature Photonics*, 6(11):773–776, November 2012.

[9] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum Computation by Adiabatic Evolution. *arXiv:quant-ph/0001106*, January 2000. arXiv: quant-ph/0001106.

[10] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev. Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation. *SIAM Journal on Computing*, 37(1):166–194, January 2007.

[11] Arnab Das and Bikas K. Chakrabarti. Quantum Annealing and Analog Quantum Computation. *Reviews of Modern Physics*, 80(3):1061–1081, September 2008. arXiv: 0801.2193.

[12] Giuseppe E. Santoro and Erio Tosatti. Optimization using quantum mechanics: quantum annealing through adiabatic evolution. *Journal of Physics A: Mathematical and General*, 39(36):R393, 2006.

[13] Gernot Schaller and Ralf Schützhold. The role of symmetries in adiabatic quantum algorithms. *Quantum Info. Comput.*, 10(1):109–140, January 2010.

[14] 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), November 2008. arXiv: 0808.1935.

[15] Nanyang Xu, Jing Zhu, Dawei Lu, Xianyi Zhou, Xinhua Peng, and Jiangfeng Du. Quantum Factorization of 143 on a Dipolar-Coupling NMR system. *Physical Review Letters*, 109(26), December 2012. arXiv: 1111.3726.

[16] Raouf Dridi and Hedayat Alghassi. Prime factorization using quantum annealing and computational algebraic geometry. *Scientific Reports*, 7:43048, February 2017.

[17] Franz Schwabl. *Quantenmechanik (QM I): eine Einführung ; mit 16 Tabellen und 127 Aufgaben*. Springer-Lehrbuch. Springer, Berlin, 6. aufl., korr. nachdr edition, 2005. OCLC: 179848502.

[18] M. Born and V. Fock. Beweis des Adiabatensatzes. *Zeitschrift für Physik*, 51(3-4):165–180, March 1928.

[19] Albert Messiah. *Quantum mechanics*. Dover Publ., Mineola, NY, 2014. OCLC: 931594080.

[20] David J. Griffiths. *Quantenmechanik: Lehr- und Übungsbuch*. ph - Physik. Pearson, München Harlow Amsterdam Madrid Boston San Francisco Don Mills Mexico City Sydney, 2., aktualisierte auflage edition, 2012. OCLC: 844999804.

[21] Andrew Das Arulsamy. Accuracy of the quantum adiabatic theorem in its original form. *arXiv:0805.0350 [physics]*, May 2008. arXiv: 0805.0350.

[22] Amar C. Vutha. A simple approach to the Landau-Zener formula. *European Journal of Physics*, 31(2):389–392, March 2010. arXiv: 1001.3322.

[23] C. J. Foot. *Atomic physics*. Number 7. Atomic, Optical, and laser physics in Oxford master series in physics. Oxford University Press, Oxford ; New York, 2005. OCLC: ocm57478010.

[24] Edwin Kah Pin Chong and Stanislaw H. Żak. *An introduction to optimization*. Wiley-Interscience series in discrete mathematics and optimization. Wiley-Interscience, Hoboken, N.J, 3rd ed edition, 2008. OCLC: ocn171049699.

[25] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem. *Science*, 292(5516):472–475, April 2001. arXiv: quant-ph/0104129.

[26] Hans Riesel. *Prime numbers and computer methods for factorization*. Number 57 in Progress in mathematics. Birkhäuser, Boston, rev. and corr. 2. print edition, 1987. OCLC: 20770981.

[27] Chris J.C. Burges. Factoring as Optimization. Technical report, January 2002.

[28] Sebastian Krämer, David Plankensteiner, Laurin Ostermann, and Helmut Ritsch. QuantumOptics.jl: A Julia framework for simulating open quantum systems. *Computer Physics Communications*, 227:109 – 116, 2018.