A Comprehensive Description of the Quantum HHL Algorithm and its Application in the Cryptanalysis of the AES

valentinpi[◦]
Student Number: Redacted
Date of Birth: Redacted

Bachelor Thesis
Bachelor of Science
Major in Computer Science

First Supervisor: Redacted
Second Supervisor: Redacted
Freie Universität Berlin, Institute for Computer Science

Date of Submission: January 23, 2023

[◦]E-Mail: valenpi@gmx.de - Website: valentinpi.github.io

Abstract

Systems of linear equations appear almost everywhere in the mathematical sciences. Let it be machine learning, economic simulations, geometry or even in the field of cryptography. It is commonly known, that the classical Gaussian elimination method achieves a runtime of $\mathcal{O}(N^3)$ for such a system of $N \in \mathbb{N}_{\geq 1}$ equations and variables. The fastest known classical approximation algorithm, the so-called conjugate gradient method, yields a complexity of $\mathcal{O}(Ns\sqrt{\kappa}\log_2(1/\varepsilon))$, where $\kappa \in \mathbb{R}_{\geq 1}$ is the condition number of the matrix, $s \in \mathbb{N}$ its sparsity and $\varepsilon > 0$ the error cap.

In this thesis, we will give a description and a mathematically rigorous analysis of the quantum algorithm by Harrow, Hassidim and Lloyd (HHL), which achieves an exponential speedup to a solution to this problem given several restrictions, to a runtime of about $\tilde{\mathcal{O}}(\kappa^2 s^4 \log_2(N)/\varepsilon)$. We further discuss its improvements and limitations. Our contribution lies in the explicit description of the smaller auxiliary algorithms involved, as well as more detailled runtime and error bounds.

Lastly, we describe, how to create simple systems of equations for key recovery of AES encrypted blocks and shortly present recent results on the application of HHL for the cryptanalysis of the Advanced Encryption System (AES) block cipher.

Zusammenfassung

Lineare Gleichungssysteme lassen sich an fast jeder Stelle in den mathematischen Wissenschaften wiederfinden. Sei es im maschinellen Lernen, ökonomischen Simulationen, in der Geometrie oder in der Kryptographie. Es ist im Allgemeinen bekannt, dass die klassische Lösungsmethode durch Gaußsche Eliminierung eine Laufzeit von $\mathcal{O}(N^3)$ für ein System von $N \in \mathbb{N}_{\geq 1}$ Gleichungen und Variablen besitzt. Der schnellste bekannte klassische Approximationsalgorithmus, die sogenannte Conjugate Gradient-Methode, besitzt eine Komplexität von $\mathcal{O}(Ns\sqrt{\kappa}\log_2(1/\varepsilon))$, wobei $\kappa \in \mathbb{R}_{\geq 1}$ die Konditionsnummer der Matrix, $s \in \mathbb{N}$ die maximale Anzahl der Einträge pro Zeile und $\varepsilon > 0$ die erlaubte Fehlerschranke ist.

In dieser Arbeit geben wir eine Beschreibung und eine mathematisch rigorose Analyse von dem Quantenalgorithmus von Harrow, Hassidim und Lloyd (HHL), welcher für die Lösung eines linearen Gleichungssystemes eine exponentielle Beschleunigung, unter mehreren Einschränkungen, zu einer Laufzeit von etwa $\tilde{\mathcal{O}}(\kappa^2 s^4 \log_2(N)/\varepsilon)$ erreicht. Wir diskutieren weiterhin die Verbesserungen und Einschränkungen von dem Algorithmus. Unser Beitrag liegt in der expliziten Beschreibung der kleineren Hilfsalgorithmen, welche involviert sind, sowie detailliertere Laufzeit- und Fehlerschranken.

Zuletzt beschreiben wir, wie einfache Gleichungssysteme für die Schlüsselgewinnung aus mit AES verschlüsselten Datenblöcken formuliert werden können und präsentieren außerdem kurz neue Ergebnisse in der Anwendung von HHL für die Kryptanalyse von dem Advanced Encryption System (AES) Blockchiffre.

Selbstständigkeitserklärung

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Bachelorarbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch bei keiner anderen Universität als Prüfungsleistung eingereicht.

Datum: _____ Unterschrift: _____

Contents

List of Algorithms

List of Figures

List of Tables

List of Abbreviations

Abbreviation	**Full Form**
AA	*Amplitude Amplification*
AES	*Advanced Encryption Standard*
BCD	*Binary Coded Decimal*
eq.	*equation*
et al.	*and others* (Latin: et alia)
i.e.	*that is* (Latin: id est)
iff	*if and only if*
LCU	*linear combination of unitaries*
LSB	*least significant bit*
MSB	*most significant bit*
NIST	*National Institute of Standards and Technology*
poset	*partially-ordered set*
QM	*Quantum Mechanics*
s.t.	*such that*
SLE	*System of Linear Equations*
SVD	*Singular Value Decomposition*
VTAA	*Variable Time Amplitude Amplification*
wlog.	*without loss of generality*
wrt.	*with respect to*

List of Notations

Let $m, n, q \in \mathbb{N}_{\geq 1}$ here, if not said otherwise. The following meanings for the symbols are used, if no other definition is specified.

Notation	Explanation
$\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$	The sets of natural, integral, rational, real and complex numbers. $0 \in \mathbb{N}$ here.
M_P	For M a set and P a logical predicate over M, the set $\{a \in M \mid P(a)\}$. For instance, $\mathbb{N}_{\geq 1}$.
$\rightsquigarrow$	Informal notation for an implication.
$x \circ M$	If $x \in U$ for some universe U and $M \subseteq U$ and $\circ \colon U \times U \mapsto U$, the set $\{x \circ y \mid y \in M\}$
$\mathrm{Im}(f)$	For a function $f \colon A \to B$ with A and B being sets, the image $f(A)$.
$\ker(f)$	For a function $f \colon A \to \mathbb{C}$ with A being a set, the preimage of zero $f^{-1}(0)$, i.e. the kernel.
$\otimes$	*Kronecker product*, the standard tensor product used here.
$\simeq$	Isomorphy relation.
$\cong$	Isometric isomorphy equivalence relation.
$\hookrightarrow\!\!\!\!\rightarrow$	Mapped under isomorphism.
$\overset{\cong}{\hookrightarrow\!\!\!\!\rightarrow}$	Mapped under isometric isomorphism.
$[a, b]_M$	For a poset $(M, \leq)$ and $a, b \in M$, the set $\{r \in M \mid a \leq r \leq b\}$.[2]
id_M	For a set M, the identity function $\mathrm{id}_M \colon M \to M, x \mapsto x$.
$A^{\otimes n}$	For $p \in \mathbb{N}_{\geq 1}$ and some $A \in \mathbb{C}^{m \times p}$, the tensor product power $\bigotimes_n A$.
$A^*, A^t, A^\dagger$	For a matrix $A \in \mathbb{C}^{m \times n}$, the associated conjugate, transposed and adjoint matrices.
$\mathbb{F}^{m \times n}$	Set of matrices of format $m \times n$ with coefficients from a field $\mathbb{F}$.
$\mathbb{F}_q$	The set $[0, q-1]_{\mathbb{N}}$ for $q \in \mathbb{N}_{\geq 1}$.
$\mathrm{GF}(p)$	*Galois field* with p elements, where $p \in \mathbb{N}$ is prime.
δ_{ij}	*Kronecker delta* for $i, j \in \mathbb{N}$. Defined as $\delta_{ij} \coloneqq (i = j)$.
E_n	Unit matrix of size $n \times n$.
σ_y	*Pauli Y matrix* $-i \left\|1\right\rangle \left\langle 0\right\| + i \left\|0\right\rangle \left\langle 1\right\|$ [1, p. 168].
$\left\|k\right\rangle, k \in \mathbb{N}$	kth canonical basis vector of the Hilbert space $\mathbb{C}^n$, where $k < n$.
rk	Matrix rank.
$\theta_{\mathcal{F}}, \theta \in \mathbb{N}$	$\theta_{\mathcal{F}} \in \mathbb{R}$ is the real number represented by θ in a floating-point-format. For instance, in IEEE-754, a fixed-point-format or BCD.
$\left\|\lambda\right\rangle, \lambda \in \mathbb{R}$	$\left\|\lambda\right\rangle$ denotes a canonical basis vector $\left\|k\right\rangle$ of $\mathbb{C}^n$, $k \in [0, n-1]_{\mathbb{N}}$, s.t. $k_{\mathcal{F}}$ is close to λ. When we develop a unitary, that utilizes a λ value, the necessary conversions are implicitely assumed to be performed.
$R[x_1, ..., x_n]$	The ring of polynomials with coefficients from a ring R over the variable symbols $x_1, ..., x_n$.
S^{n-1}	The sphere $\{x \in \mathbb{R}^n \mid \|x\| = 1\}$ with the standard norm.
$\sphericalangle(\cdot, \cdot)$	Angle between two vectors in a euclidian vector space $(V, \langle \cdot, \cdot \rangle)$. Defined as $\sphericalangle(u, v) \coloneqq \arccos\left(\frac{\langle u, v \rangle}{\|u\| \|v\|}\right)$ for $u, v \in V \setminus \{0\}$ with $\|\cdot\|$ being the associated norm.
ω_N	$\omega_N \coloneqq e^{2\pi i / N}$. Looking in a mathematically positive direction on S^1, the *first Nth complex unit root besides* 1.
$\mathrm{diag}(A_1, ..., A_r)$	For $A_1, \in \mathbb{F}^{m_1 \times n_1}, ..., A_r \in \mathbb{F}^{m_r \times n_r}$ for a field $\mathbb{F}$ and $r, m_1, n_1, ..., m_r, n_r \in \mathbb{N}_{\geq 1}$, the diagonal matrix with blocks $A_1, ..., A_r$.
$U^\perp$	For a subspace $U \subseteq V$ of a vector space V, the *orthogonal complement* of U.
$\mathrm{E}[X]$	For a discrete finite random variable $X \colon (\Omega, \Pr) \to \mathbb{R}$ over a discrete finite probability space $(\Omega, \Pr)$, its expectation value $\sum_{x \in \Omega} \Pr(X = x) x$.
$\mathrm{poly}(T_1, ..., T_n)$	With runtime terms $T_1, ..., T_n$, for one $p \in \mathbb{R}[x]$, which depends only on $T_1, ..., T_n$, the class $\mathcal{O}(p)$.
$A \leq_p B$	A is polynomially time reducible to B, see Appendix C, for languages $A, B \subseteq \Sigma^*$.

Indexing of Vectors and Matrices Vectors in $\mathbb{F}^n$, where $\mathbb{F}$ is a field, will always be interpreted as column vectors. Vector and matrix entries are not zero-indexed, if not said otherwise. Matrices are indexed column-first. We may also index complex numbers, since $\mathbb{C} \cong \mathbb{R}^2$. The bra-ket notation is used for valid quantum registers, i.e. normalized vectors in Hilbert spaces and their associated functionals. Otherwise not. We never omit the bra-ket notation to index the vectors. We further use a notation for generating matrices of form $A \coloneqq (p(i,j))_{i,j \in m \times n} \in \mathbb{F}^{m \times n}$. By that, we mean that for any $i \in [1,m]_{\mathbb{N}}$, $j \in [1,n]_{\mathbb{N}}$, we have $a_{ij} = p(i,j)$, where $p \colon [1,m]_{\mathbb{N}} \times [1,n]_{\mathbb{N}} \to \mathbb{F}$ is a function.

Standard Product and Norm We use the general definition from [2, p. 219] for standard products. The symbols $\langle \cdot | \cdot \rangle$ and $\| \cdot \|$ are reserved for the complex standard product and its induced norm, defined as:

$$\langle u|v\rangle \coloneqq \sum_{i=1}^{n} u_i v_i^* \qquad \|u\| \coloneqq \sqrt{\langle u,u\rangle} \tag{0.0.1}$$

For $|u\rangle, |v\rangle \in \mathbb{C}^n$. Furthermore, the symbol $\| \cdot \|$ is also reserved for the operator norm used here, see Theorem 2.12.

Sets and Operations When considerung a group, ring, field, vector space or another structure, we often omit the explicit statement of the associated operations.

Switching between Matrices and Tuples Note, that by *column-major* and *row-major*, we refer to the order, in which the entries of a tuple or matrix are mapped to a respective matrix or tuple. For instance, we may enumerate the vector $(0,1,2,3) \in \mathbb{R}^4$ in either column-major- or row-major-enumeration into a 2×2-matrix, yielding respectively:

$$\begin{pmatrix} 0 & 2 \\ 1 & 3 \end{pmatrix} \text{ or } \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix} \tag{0.0.2}$$

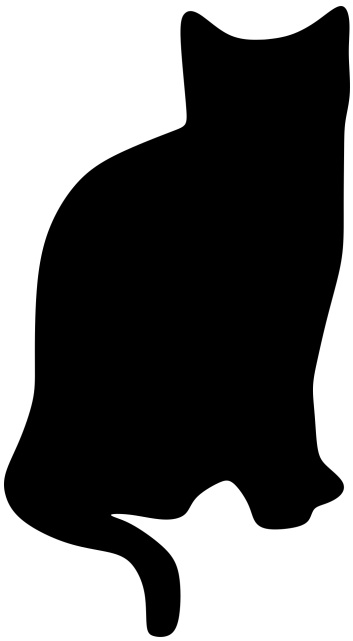


Figure 1: A famous cat. She is cute and not a sign of bad luck. She is both completely blacked out with no life sign, whilst standing upright.

1 Introduction

The main goal of this thesis is to present the quantum algorithm for solving systems of linear equations proposed by Harrow et al. [3] in 2008 in full detail and in the original formulation. We then apply it to the cryptanalysis of AES. For that, we shortly present the results by Chen [4] and Ding [5].
This and the next section are dedicated to providing the foundations to this thesis. Note that, although this thesis is written in English, we will partly give references to *German* standard literature. The focus is to present these results following rigorous mathematical sources.
This thesis is divided into four sections. In Section 1, we first introduce necessary mathematical background knowledge on Hermitian matrices, matrix invertibility criteria and polynomial factor rings. Section 2 presents multiple auxiliary quantum algorithms, including Qutrit Rotation, Amplitude Amplification, Quantum Phase Estimation and Hamiltonian Simulation. Section 3 then presents a full, rigorous description of the original HHL algorithm, as described by Harrow et al.. Section 4 closes the thesis by introducing the AES cipher and linearization techniques, as well as presenting the current state of the art of the approach.

1.1 Background Knowledge in Quantum Computation

Classical Computers, with which nowadays we are all familiar with, utilize the notion of a logical *bit* to process information. In the simplest case, a bit is physically implemented by a small transistor, capable of storing an electronic current. This allows the physical machine to differentiate between the logical values 0 and 1 and is the foundation of all other activities in a classical electronic computer.

Quantum Mechanics is a physical theory of microscopically small particles. Such particles exhibit many interesting properties, such as the so-called *particle-wave duality* [1, pp. 4-8]. We model quantum particles as elements of Hilbert spaces, of which we can measure some properties. Such measurable properties are called *observables*. A particle is always in a state, which, in turn, is a superposition of several special states. Let us dive into a little more detail.

For that, we follow [6, pp. 29-39]. Suppose $t_0 \in \mathbb{R}_{\geq 0}$ is the starting point of our investigation of a very small particle, take it to be an electron or a photon, which is in a state $|\psi(t)\rangle$ at time point $t \in [t_0, \infty)$. The state is an element of a Hilbert space $\mathcal{H}$. Especially, $|\psi(\cdot)\rangle : [t_0, \infty) \to \mathcal{H}$ is thus the map capturing the development of the state over time. Let us assume $\mathcal{H} = \mathbb{C}^2$ for convenience for the partial derivative below. There are general operator derivatives [2, p. 126 ff.], but here, we will not get into that. In Borns statistical interpretation of the quantum mechanical wave function, $\||\psi(t)\rangle\| = 1$ must hold at all times [1, pp. 3–5], aligning with the stochastical nature of quantum particles. The particle state especially obeys the following version of the fixed-position, time-dependent *Schroedinger equation* [6, p. 38]:

$$i\frac{\partial}{\partial t}|\psi(t)\rangle = H(t)\,|\psi(t)\rangle \tag{1.1.1}$$

H is an operator, called the *Hamiltonian* of the particle, and represents its observable energy with its eigenvalues. It is often the sum of potential and kinetic energy, each also being represented by an operator. Furthermore, the *time postulate* [6, p. 38] holds in the theory, stating that there is a map $U\colon [t_0, \infty) \to \{O\colon \mathcal{H} \to \mathcal{H} \mid O \text{ is a unitary operator}\}$, satisfying:

$$|\psi(t)\rangle = U(t)\,|\psi(t_0)\rangle \tag{1.1.2}$$

Also, $U(t_0) = \mathrm{id}_{\mathcal{H}}$ for sure. So, according to the time postulate, quantum states are only transformed unitarily. One may take this postulate to be the starting point for the idea of quantum computation.

Quantum computers differ from classical computers. Here we utilize the notion of particles, that can form superpositions of bit values. For this thesis, the physics of these systems is less interesting to us, than their computational consequences, and we will not discuss the implementation of quantum hardware. To us, a so-called *qubit* is capable of spanning a superposition between the two classical bit values 0 and 1. This system is represented by a complex vector in the Hilbert space $\mathbb{C}^2$, being the complete complex Euclidian vector space of 2-complex-component vectors. These systems, as mentioned

above, can only be transformed unitarily to us. Using the results of quantum mechanics, we hope to find more efficient algorithms for solving tough computational problems. It has been shown by authors such as Deutsch, Jozsa, Bernstein, Vazirani, Grover and Shor, that quantum computers, for some special problems, are indeed able to produce exponential speedups to their classical counterpart algorithms [7]. It has also been shown, that quantum computers and classical computers are computationally equivalent, as classical systems can simulate quantum computers and vice versa due to Toffoli-gates [7, p. 29 f.]. This also means, that classical issues like the halting problem cannot be resolved with this new model. The complexity-theoretic landscape of classical and quantum complexity classes is much more complex. In this thesis, we will discuss an approach to solving SLEs. As in classical computation, we may also use the notion of *gates* to describe algorithms on qubits and qubit registers, with these gates corresponding to unitary matrices.

With the gate model of quantum computing, a framework was given for non-physicists to design quantum algorithms by applying unitary transformations to a quantum state. Despite that, research in the field of Quantum Computer Science is still partly dominated by terminology from QM. We will occasionally talk about Hamiltonians in this thesis, although we will not explicitely talk about energies of particles, unlike adiabatic quantum computation, for instance. This specific term is due to the above mentioned Schroedinger equation, where H is always Hermitian. Physicists use the terms *Hermitian operator* and *Hamiltonian* synonymously.

This bachelor thesis is designed to be mostly self-sufficient, besides a required background in linear algebra, analysis and quantum computational principles.

1.2 Finite-Dimensional Hermitian Operator Theory

This subsection will introduce Hermitian matrices and present some important mathematical results. Some examples will also be mentioned. Throughout this subsection, let $m, n \in \mathbb{N}_{\geq 1}$. As we will talk about quantum computing, we shall revisit the definition of unitary matrices first, after some remarks on our terminology.

Remark 1.1. We first visit a few definitions from functional analysis. Complex matrices in $\mathbb{C}^{m\times n}$ are linear maps between the vector spaces $\mathbb{C}^n$ and $\mathbb{C}^m$, which are in turn, due to the standard norm, *Banach spaces* [2, p. 2]. Especially, complex matrices are continuous [8, p. 35], meaning that we can also call them by their functional-analytic term *operators* [2, p. 49]. Operators, that map into scalar spaces, such as $\mathbb{C}$, are also called *functionals*.

Definition 1.2. An invertible matrix $U \in \mathbb{C}^{n\times n}$ is called *unitary*, if $U^{-1} = U^\dagger$.

Example 1.3. Using the exponential form of the sine and cosine functions from Definition B.2 and the trigonometric pythagoras from Theorem B.3, one may easily verify that the two-dimensional rotation by an angle $\varphi \in [0, 2\pi)$ is unitary:

$$\begin{pmatrix}\cos(\varphi) & -\sin(\varphi)\\ \sin(\varphi) & \cos(\varphi)\end{pmatrix}\begin{pmatrix}\cos(\varphi) & -\sin(\varphi)\\ \sin(\varphi) & \cos(\varphi)\end{pmatrix}^\dagger = \begin{pmatrix}\cos^2(\varphi)+\sin^2(\varphi) & 0\\ 0 & \sin^2(\varphi)+\cos^2(\varphi)\end{pmatrix}^\dagger = E_2 \quad (1.2.1)$$

Theorem 1.4. Let $U \in \mathbb{C}^{n\times n}$ with rows $u_1, ..., u_n \in \mathbb{C}^n$ and columns $v_1, ..., v_n \in \mathbb{C}^n$. The following are equivalent:

- U is unitary.
- $\{u_1, ..., u_n\}$ is an orthonormal basis of $\mathbb{C}^n$.
- $\{v_1, ..., v_n\}$ is an orthonormal basis of $\mathbb{C}^n$.

For the proof, see [9, pp. 351-352]. Remember, that unitary matrices represent steps in quantum algorithms.

Theorem 1.5. Unitary matrices are *length-preserving/isometric*, meaning that for any unitary $U \in \mathbb{C}^{n\times n}$ and $x \in \mathbb{C}^n$ it holds that $\|Ux\| = \|x\|$. Especially they preserve the standard product, meaning that for $u, v \in \mathbb{C}^n$, we have $\langle Uu, Uv\rangle = \langle u, v\rangle$.

For the proof, see [9, pp. 350-351]. We now introduce Hermitian matrices.

Definition 1.6. We call a normalized eigenvector $|v\rangle \in \mathbb{C}^n$ of a matrix $U \in \mathbb{C}^{n\times n}$ an *eigenstate*.

Definition 1.7. A matrix $H \in \mathbb{C}^{n\times n}$ is called *Hermitian*[3], if $H = H^\dagger$. We also call Hermitian matrices *Hamiltonians*.

Example 1.8. Consider the following matrix:

$$\begin{pmatrix} 1 & i \\ -i & 2 \end{pmatrix}^\dagger = \begin{pmatrix} 1 & -i \\ i & 2 \end{pmatrix}^* = \begin{pmatrix} 1 & i \\ -i & 2 \end{pmatrix} \tag{1.2.2}$$

Theorem 1.9. Every Hermitian matrix $H \in \mathbb{C}^{n\times n}$ possesses at most n eigenvalues, with all of them being real. There is an orthonormal basis of $\mathbb{C}^n$, which is composed entirely of eigenvectors of H, also called an *eigenbasis*.

For the proof see [9, pp. 360-362]. It is clear that, since eigenvectors are by definition non-zero, we can also normalize the eigenvectors mentioned to a length of one and thus obtain an orthonormal basis. In general, any basis of eigenvectors is called an eigenbasis. With the Gram-Schmidt-orthonormalization-procedure [10, p. 185] however, an orthonormal basis can be acquired algorithmically from a non-orthonormal eigenbasis. Note further, that some eigenstates may also be associated with the zero eigenvalue.

Example 1.10. The following Hermitian matrix has eigenvalues 2 and 0 with corresponding eigenvectors $|0\rangle$ and $|1\rangle$, which form an eigenbasis of $\mathbb{C}^2$:

$$\begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} \tag{1.2.3}$$

Theorem 1.11 (Spectral Decomposition)**.** Given a Hermitian $H \in \mathbb{C}^{n\times n}$ with eigenvalues $\lambda_1, ..., \lambda_n \in \mathbb{R}$ and eigenbasis $|v_1\rangle, ..., |v_n\rangle \in \mathbb{C}^n$, it holds that:

$$H = \sum_{i=1}^{n} \lambda_i |v_i\rangle \langle v_i| \tag{1.2.4}$$

Proof. It suffices to show the statement for the vectors in the eigenbasis. Since the vectors are orthogonal, we observe for any $j \in [1, n]_{\mathbb{N}}$:

$$\sum_{i=1}^{n} \lambda_i |v_i\rangle \langle v_i|v_j\rangle = \lambda_j |v_j\rangle = H |v_j\rangle \tag{1.2.5}$$

■

Corollary 1.12. If the Hermitian matrix in Theorem 1.11 is invertible, the eigenvalues are all non-zero and the spectral decomposition of H^{-1} is given by:

$$H^{-1} = \sum_{i=1}^{n} \lambda_i^{-1} |v_i\rangle \langle v_i| \tag{1.2.6}$$

Proof. We prove the first statement by contradiction. With reordering, we may assume wlog., that $\lambda_1 = 0$. Then $H|v_1\rangle = \lambda_1 v_1 = 0 = H|0\rangle$, contradicting the bijectivity of H. ↯ We observe, that $H\left(\sum_{i=1}^{n} \lambda_i^{-1} |v_i\rangle \langle v_i|\right) |v_k\rangle = \lambda_k \lambda_k^{-1} |v_k\rangle = |v_k\rangle$ for all $|v_k\rangle$ by the above formula, proving equality. ■

This theorem allows us to write a given Hermitian matrix more compactly. It can surely also be used for generally any matrix, where the eigenvalues involved may then be complex. Another useful decomposition of matrices is presented in the following.

[3] After Charles Hermite.

Theorem 1.13 (Outer Product Form of the SVD)**.** Let $A \in \mathbb{C}^{m\times n}$ and $r \coloneqq \mathrm{rk}(A)$. There are so-called *singular values* $\sigma_1, ..., \sigma_r \in \mathbb{R}_{>0}$ with $\sigma_1 \geq ... \geq \sigma_r$ and orthonormal systems, comprised of so-called *singular vectors*, $\{|u_1\rangle, ..., |u_r\rangle\} \subset \mathbb{C}^m$ and $\{|v_1\rangle, ..., |v_r\rangle\} \subset \mathbb{C}^n$, such that:

$$A = \sum_{j=1}^{r} \sigma_j |u_j\rangle \langle v_j|$$

The proof is given in [11, p. 153-157].

Corollary 1.14 (SVD). Any matrix $A \in \mathbb{C}^{m\times n}$ can be written in the form

$$A = U\Sigma V^\dagger \tag{1.2.7}$$

where $\Sigma \coloneqq \mathrm{diag}(\sigma_1, ..., \sigma_r, 0, ..., 0) \in \mathbb{C}^{m\times n}$ with $\sigma_1, ..., \sigma_r \in \mathbb{R}_{>0}$ being the singular values of A and $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ being unitary.

Proof. Consider the outer form SVD of A, see Theorem 1.13. Extend $\{|u_1\rangle, ..., |u_r\rangle\}$ and $\{|v_1\rangle, ..., |v_r\rangle\}$ to an orthonormal basis each for $\mathbb{C}^m$ and $\mathbb{C}^n$ respectively via $\{|u_1\rangle, ..., |u_m\rangle\}$ and $\{|v_1\rangle, ..., |v_n\rangle\}$. The computation

$$A = \sum_{j=1}^{r} \sigma_j |u_j\rangle \langle v_j| = \begin{pmatrix} |u_1\rangle & \cdots & |u_m\rangle \end{pmatrix} \mathrm{diag}(\sigma_1, ..., \sigma_r, 0, ..., 0) \begin{pmatrix} \langle v_1| \\ \cdots \\ \langle v_n| \end{pmatrix} =: U\Sigma V^\dagger \tag{1.2.8}$$

which we can directly verify using the matrix product gives the statement. ■

We cannot invert non-invertable matrices. The following definition gives us a different notion of invertibility.

Definition 1.15 (Moore-Penrose Pseudoinverse)**.** Let $A \in \mathbb{C}^{m\times n}$ possess the SVD $A = U\Sigma V^\dagger$ with singular values $\sigma_1, ..., \sigma_r \in \mathbb{R}_{>0}$. Then we define the *Monroe-Pense Pseudoinverse* to be

$$A^+ \coloneqq V\Sigma^+ U^\dagger \tag{1.2.9}$$

with $\Sigma^+ \coloneqq \mathrm{diag}\left(\frac{1}{\sigma_1}, ..., \frac{1}{\sigma_r}, 0, ..., 0\right)$.

This definition follows [12, pp. 41-42]. For $m = n$ and A being invertible for instance, we can verify $A^+ = A^{-1}$ via $AA^+ = U\Sigma V^\dagger V\Sigma^+ U^\dagger = E^m$.

Definition 1.16. The *matrix exponential function* is defined by:

$$\exp\colon \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}, M \mapsto \sum_{k=0}^{\infty} \frac{M^k}{k!} \tag{1.2.10}$$

We shall also note $\exp(M) =: e^M$.

Note that this series is a multidimensional limit. The following lemma gives us the convergence and two other properties.

Lemma 1.17 (Properties of the matrix exponential function). Let $M, N \in \mathbb{C}^{n\times n}$. The following holds:

(i) $\exp(0) = E_n$.

(ii) $\exp(M)$ converges.

(iii) If $MN = NM$, then $\exp(M + N) = \exp(M)\exp(N)$.

The proof can be found in [13, p. 9]. The previous statements and definitions are generalizations of known facts from the study of euclidian/unitarian vector spaces. Now we want to study the problem of generating a unitarian matrix with a Hermitian matrix.

Theorem 1.18. For any Hermitian matrix $H \in \mathbb{C}^{n\times n}$ and $t \in \mathbb{R}$, e^{iHt} is unitary.

Remark 1.19. The parameter t is introduced, as the unitary described may be interpreted as a time evolution of a particle, as described in the introduction.

We shall demonstrate the notion of the matrix exponential by giving a proof to this statement. Without proof, note that taking the transpose of a matrix and taking the conjugate are both continuous mappings, meaning that we can move these operations inside of the matrix exponential series.

Proof. In the following, we move the adjunction inside of the series. Since taking the adjoint is compatible both with addition and multiplication in each matrix entry, it holds:

$$\left(e^{iHt}\right)^\dagger = \sum_{k=0}^{\infty} \frac{(-1)^k i^k (H^\dagger)^k t^k}{k!} = e^{-iH^\dagger t} = e^{-iHt} \tag{1.2.11}$$

Since $H = H^\dagger$ and thus $HH^\dagger = H^\dagger H$, we can use Lemma 1.17 and conclude:

$$e^{iHt}\left(e^{iHt}\right)^\dagger = e^{iHt-iHt} = e^0 = E_n \tag{1.2.12}$$

■

Theorem 1.20. Suppose $U \in \mathbb{C}^{n\times n}$ is unitary with eigenvalue $\lambda \in \mathbb{C}$. Then there is a number $\theta \in [0, 1)$ with $\lambda = e^{i2\pi\theta}$, called the *phase* of the eigenvalue.

Proof. It suffices to show that the magnitude of the eigenvalue is 1. Let v be an eigenvector to λ. With Theorem 1.5, we have $\|Uv\| = \|\lambda v\| = |\lambda|\|v\| = \|v\|$ and $|\lambda| = 1$, since by definition $v \neq 0$. ■

Theorem 1.21. If a Hermitian matrix $H \in \mathbb{C}^{n\times n}$ has eigenvalue λ with eigenvector v, then the associated unitary matrix exponential $e^{iHt}, t \in \mathbb{R}$, has eigenvalue $e^{i\lambda t}$ with eigenvector v.

Proof. We have

$$e^{iHt}v = \sum_{k=0}^{\infty} \frac{i^k t^k}{k!} H^k v = \sum_{k=0}^{\infty} \frac{i^k \lambda^k t^k}{k!} v = e^{i\lambda t} v \tag{1.2.13}$$

■

Remark 1.22. This theorem shows that an eigenbasis of e^{iHt} is given by an eigenbasis of H. It is important to note, that, due to Theorem 1.11 and Theorem 1.21, we can write the spectral decomposition of e^{iHt} as:

$$e^{iHt} = \sum_{i=1}^{n} e^{i\lambda_i t} |v_i\rangle \langle v_i| \tag{1.2.14}$$

The proof is analogous to the one of Theorem 1.11.

1.3 Matrix Condition Number and Sparsity

The main tool for quantifying the toughness of a matrix invertibility problem is the *condition number*. There are multiple ways of defining the condition number, we use the following definition following Lyche [11]

Definition 1.23. The *condition number* $\kappa(A) \in \mathbb{R}_{\geq 1}$ of a matrix $A \in \mathbb{C}^{m\times n}$ with singular values $\sigma_1, ..., \sigma_r \in \mathbb{R}_{>0}$, $r \coloneqq \operatorname{rank}(A)$ is defined by

$$\kappa(A) \coloneqq \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}, \text{ where } \sigma_{\max}(A) \coloneqq \max\{\sigma_1, ..., \sigma_r\}, \sigma_{\min}(A) \coloneqq \min\{\sigma_1, ..., \sigma_r\} \tag{1.3.1}$$

Furthermore, we set for $m = n$, A invertible and, possibly duplicate, eigenvalues $\lambda_1, ..., \lambda_n \in \mathbb{C} \setminus \{0\}$ of A the condition number as

$$\kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}, \text{ where } \lambda_{\max}(A) \coloneqq \max\{|\lambda_1|, ..., |\lambda_n|\}, \lambda_{\min}(A) \coloneqq \min\{|\lambda_1|, ..., |\lambda_n|\} \tag{1.3.2}$$

We can also set $\kappa(A) \coloneqq \|A\|\|A^+\|$ using the Moore-Penrose pseudoinverse, see Definition 1.15, so the concrete use of condition numbers depends on the current context.

Remark 1.24. Note, that

- $\kappa(A) \geq 1$ always holds, due to $0 < \sigma_{\min}(A) \leq \sigma_{\max}(A)$ and for the second part of the definition analogously. Especially, $0 < \frac{1}{\kappa(A)} \leq 1$.
- if $\kappa(A)$ is very large, then we call A *ill-conditioned*.

Example 1.25. Consider a diagonal matrix D with diagonal elements $d_{11}, ..., d_{nn} \in \mathbb{C}_{\neq 0}$. Then $\kappa(D) = \max_{i\in[1,n]_\mathbb{N}} |d_{ii}| / \min_{i\in[1,n]_\mathbb{N}} |d_{ii}|$, which allows us to increase the condition arbitrarily. Consider for instance for $n \geq 2$ and $j \in \mathbb{N}$ the matrix $D \coloneqq 2^j |0\rangle \langle 0| + \sum_{i=1}^{n-2} |i\rangle \langle i| + 2^{-j} |n-1\rangle \langle n-1|$. One may ask the question, whether there are non-trivial ill-conditioned matrices.

Example 1.26. One particularly interesting class of examples are *Hilbert matrices*, where the *n-th Hilbert matrix* is defined as

$$\mathcal{H}_n \coloneqq \left(\frac{1}{i+j-1}\right)_{i,j\in n\times n} \tag{1.3.3}$$

This construction solves the question from Example 1.25: $\mathcal{H}_n$ is clearly Hermitian and it can also be shown, that it is invertible by explicitely giving the inverse as in [14, pp. 302, 306]. [15, p. 51] gives the bound $\lambda_{\min}(\mathcal{H}_n) \in \Theta(\sqrt{n}(1+\sqrt{2})^{-4n})$ and, following the result cited in [16, p. 111], we also have $\lambda_{\max}(\mathcal{H}_n) \in \Theta(\pi)$. So

$$\kappa(\mathcal{H}_n) \in \Theta\left(\frac{(1+\sqrt{2})^{4n}}{\sqrt{n}}\right) \tag{1.3.4}$$

which gives the statement that this matrix is very ill-conditioned.

Definition 1.27. A matrix $A \in \mathbb{C}^{m\times n}$ is called *s-sparse*, with $s \in \mathbb{N}$, if there are at most s many non-zero entries per row or column. A is called *efficiently row-computable*, if there is an algorithm, that, for a given row or column index respectively, computes the corresponding indices of the non-zero entries in time $\mathcal{O}(s)$.

Definition 1.28. We call an invertible, Hermitian, positive-semidefinite, sparse, efficiently row-computable matrix with condition number $\kappa \in \mathbb{R}_{\geq 1}$ and for all eigenvalues $\lambda \in \mathbb{R}_{>0}$, that

$$\frac{1}{\kappa} \leq \lambda \leq 1 \tag{1.3.5}$$

well-conditioned.

1.4 Finite Polynomial Fields

This subsection is dedicated to presenting fields of polynomials, which are formed over finite fields. We use the book by Fischer [17] for the necessary algebra. Recall the formal details of a group [17, p. 5], a ring [17, pp. 171-172], a field [17, p. 174], polynomial rings [17, pp. 183-186], an ideal and generating an ideal [17, p. 206].

Definition 1.29. Let K be a field and $p \in K[x]$. The *factor ring* $K[X]/(p)$ is composed of the set $\{q + (p) \mid q \in K[X]\}$ with the operations $(q + (p)) + (q' + (p)) \coloneqq (q + q') + (p)$ and $(q + (p)) \cdot (q' + (p)) \coloneqq q \cdot q' + (p)$ for $q, q' \in K[X]$.

Here, $(p) = \{qp \mid q \in K[x]\}$ denotes the ideal generated by p. For further information and a more precise description with proof, see [17, p. 208]. Here, we use the common calculation techniques for taking modulos with polynomials via polynomial division, as also described in [17, p. 188]. Furthermore, as taking the modulo is unique, we may choose representants of the elements in $K[x]/(p)$ via the condition $\deg(q) < \deg(p)$. This, with an additional result, gives the following result.

Theorem 1.30. If $p \in K[x]$ is irreducible, then $K[x]/(p)$ is a field. If K is finite, then $|K[x]/(p)| = |K|^{\deg(p)}$.

For the proof of the first part of the statement, we refer to [17, p. 313]. For the second part, consider

$$\{q + (p) \mid q \in K[x]\} = \{q + (p) \mid q \in K[x] \wedge \deg(q) < \deg(p)\} \tag{1.4.1}$$

via the uniqueness of polynomial division.

Corollary 1.31. The factor rings $\mathrm{GF}(2)[x]/(x^8 + x^4 + x^3 + x + 1) \cong \mathbb{F}_{2^8}$ and $\mathbb{F}_{2^8}[x]/(x^4 + 1) \cong \mathbb{F}_{2^8}^4$ are fields.

Note, that GF(2) is a field because 2 is prime. Due to the isomorphism to $\mathbb{F}_{2^8}$, it formally makes sense to speak of $\mathrm{GF}(2^8)$ as a field, although 2^8 is not prime. Another factor ring that will be of interest later on is .

Example 1.32. We shall give a short example for polynomial multiplication in finite fields and the matrix representation of a multiplication with a fixed polynomial. Consider the field $\mathrm{GF}(2)^8[x]/(x^4 + 1)$. We have for instance

$$(3x^3 + x^2 + x + 2) \cdot x^3 \bmod (x^4 + 1) = 2x^3 + 3x^2 + x + 1 \tag{1.4.2}$$

The multiplication to obtain the result here is done via polynomial division in $\mathbb{R}[x]$. We obtain

$$\begin{array}{r}
\big(\quad 3x^6 + x^5 + x^4 + 2x^3\big) : \big(x^4 + 1\big) = 3x^2 + x + 1 + \dfrac{2x^3 - 3x^2 - x - 1}{x^4 + 1} \\
\underline{-\,3x^6 \qquad\qquad\quad - 3x^2} \\
\qquad\quad x^5 + x^4 + 2x^3 - 3x^2 \\
\qquad\underline{-\,x^5 \qquad\qquad\quad - x} \\
\qquad\qquad\quad x^4 + 2x^3 - 3x^2 - x \\
\qquad\qquad\underline{-\,x^4 \qquad\qquad\quad - 1} \\
\qquad\qquad\quad 2x^3 - 3x^2 - x - 1
\end{array} \tag{1.4.3}$$

Since $-3 = 3$ and $-1 = 1$ in $\mathrm{GF}(2)^8$, we have the result. Especially, as polynomial multiplication is linear, we can even form a matrix to compute these results faster. It suffices to compute the product with $\{x^3, x^2, x, 1\}$. In this case, the matrix is exactly

$$\begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix} \tag{1.4.4}$$

The coefficients for x^3, which can be found in the first row, were computed above.

2 Extensions of the Common Quantum Algorithmic Toolbox

The HHL algorithm requires the reader to have a rather large amount of preliminary knowledge. We shall introduce a set of common tools and their current state of the art, in the same sense as Barak [18, p. 415] referred to the *quantum algorithmic toolbox*. Let $n \in \mathbb{N}_{\geq 1}$ throughout this section. Recall some the common gates, that current books [6, 7, 19] on quantum computer science using the unitary gate model of quantum computation present:

$$E_N \coloneqq (\delta_{ij})_{i,j \in N \times N} \qquad \mathrm{QFT}_N \coloneqq \left(\omega_N^{(i-1)(j-1)}\right)_{i,j \in N \times N} \qquad H \coloneqq \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad \mathrm{NOT} \coloneqq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{2.0.1}$$

Here is $N \coloneqq 2^n$. Recall, that *classical functions* can be efficiently simulated using a unitary of form $|x\rangle |y\rangle \mapsto |x\rangle |y \oplus f(x)\rangle$ for a function $f \colon \mathbb{F}_2^m \to \mathbb{F}_2^n$ with $m \in \mathbb{N}_{\geq 1}$. Recall the concept of *entangled states* from [7, p. 95-96].

Remark 2.1. When we present a quantum algorithm using an algorithm description, we normally write register tensor product terms, for instance $|\mu\rangle |\nu\rangle = |\mu\rangle \otimes |\nu\rangle$, where $|\mu\rangle$ and $|\nu\rangle$ are some quantum states. Our steps can lead to an entanglement of the registers, deeming this notation to be false statements, but we shall ignore that for convenience. One may imagine a step as a application of a single large unitary, that affects all states involved.

2.1 Auxiliary Gates

Swapping Qubits

When designing a quantum algorithm, one often needs to append auxiliary qubits for other calculations. It is often not clear, whether we can discard the auxiliary qubits afterwards. One necessary requirement for that is, that our current working state is not entangled with the auxiliary state. We call the process of preparing an auxiliary state for removal *uncomputing*. The following gate assists us in that task:

Definition 2.2. The unitary SWAP-gate is defined by:

$$\mathrm{SWAP} \colon \mathbb{C}^2 \to \mathbb{C}^2, |x\rangle |y\rangle \mapsto \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} |x\rangle |y\rangle = |y\rangle |x\rangle \tag{2.1.1}$$

One may quickly observe the unitarity and correctness. By definition, the gate acts on constantly many qubits and is thus local and efficiently implementable. Successive uses of this gate allow us to uncompute multiple qubits.

Rotating Qubits

We present lemma 4 at [12, p. 25], with which we now introduce a quantum gate for the so-called *controlled rotation* of a qubit. We can imagine that as rotating the unit vector $|0\rangle$ by some angle in the Gaussian plane, see Figure 2. The mentioned paper cites this theorem for a fixed angle, but it is clear from the proof, that this can be generalized for every d-bit represented real number, $d \in \mathbb{N}_{\geq 1}$.
We will now use our notations for floating-point-values. In case of a format like IEEE-754, the proof of the efficient implementability of the following theorem may be a bit, perhaps in form of quite a few more qubits, harder[4].

Lemma 2.3 (Controlled Rotation). For a fixed $d \in \mathbb{N}_{\geq 1}$, there is an with $\mathcal{O}(d)$ local gates efficiently implementable unitary that achieves for d-bit representations of angles θ:

$$\mathrm{CR}_d \colon \mathbb{C}^{2^{d+1}} \to \mathbb{C}^{2^{d+1}}, |\theta\rangle |0\rangle \mapsto |\theta\rangle \left(\cos(\theta_{\mathcal{F}}) |0\rangle + \sin(\theta_{\mathcal{F}}) |1\rangle\right) \tag{2.1.2}$$

We call it the gate for *controlled rotations*.

[4] Including special cases like infinity or NaN, of course.

Proof. We give a detailled version of the proof in the paper of Dervovic et. al.. Note that there is one slight subtilty: The unitary that is given in the paper is not fully correct. In the unitary, we must denote the real number, that is represented by the finite-bit representation. Let

$$\mathrm{CR}_d \coloneqq \sum_{\theta\in\mathbb{F}_2^d} |\theta\rangle\langle\theta| \otimes \exp(-i\theta_{\mathcal{F}}\sigma_y) = \begin{pmatrix} e^{-i\cdot 0_{\mathcal{F}}\cdot\sigma_y} & 0 & \dots & 0 \\ 0 & e^{-i\cdot 1_{\mathcal{F}}\cdot\sigma_y} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & e^{-i\cdot(2^d-1)_{\mathcal{F}}\cdot\sigma_y} \end{pmatrix} \tag{2.1.3}$$

Note that, in the definition, θ is a bitvector and interpreted as its associated natural number. For any $|\theta\rangle \in \mathbb{F}_2^d$, it holds that:

$$\mathrm{CR}_d |\theta\rangle |0\rangle = |\theta\rangle \exp(-i\theta_{\mathcal{F}}\sigma_y)|0\rangle = |\theta\rangle \exp\begin{pmatrix} 0 & -\theta_{\mathcal{F}} \\ \theta_{\mathcal{F}} & 0 \end{pmatrix} |0\rangle \overset{(1)}{=} |\theta\rangle (\cos(\theta_{\mathcal{F}})|0\rangle + \sin(\theta_{\mathcal{F}})|1\rangle) \tag{2.1.4}$$

(1) We use Definition 1.16 and Theorem B.1 to obtain

$$\exp\begin{pmatrix} 0 & -\theta_{\mathcal{F}} \\ \theta_{\mathcal{F}} & 0 \end{pmatrix} = \sum_{k=0}^{\infty} \frac{1}{k!} \begin{pmatrix} 0 & -\theta_{\mathcal{F}} \\ \theta_{\mathcal{F}} & 0 \end{pmatrix}^k \tag{2.1.5}$$

$$= \sum_{k=0}^{\infty} \frac{1}{(2k)!} \begin{pmatrix} 0 & -\theta_{\mathcal{F}} \\ \theta_{\mathcal{F}} & 0 \end{pmatrix}^{2k} + \sum_{k=0}^{\infty} \frac{1}{(2k+1)!} \begin{pmatrix} 0 & -\theta_{\mathcal{F}} \\ \theta_{\mathcal{F}} & 0 \end{pmatrix}^{2k+1} \tag{2.1.6}$$

$$= \sum_{k=0}^{\infty} \frac{1}{(2k)!} \begin{pmatrix} (-1)^k\theta_{\mathcal{F}}^{2k} & 0 \\ 0 & (-1)^k\theta_{\mathcal{F}}^{2k} \end{pmatrix} + \sum_{k=0}^{\infty} \frac{1}{(2k+1)!} \begin{pmatrix} 0 & -(-1)^k\theta_{\mathcal{F}}^{2k+1} \\ (-1)^k\theta_{\mathcal{F}}^{2k+1} & 0 \end{pmatrix} \tag{2.1.7}$$

$$= \begin{pmatrix} \cos(\theta_{\mathcal{F}}) & -\sin(\theta_{\mathcal{F}}) \\ \sin(\theta_{\mathcal{F}}) & \cos(\theta_{\mathcal{F}}) \end{pmatrix} \tag{2.1.8}$$

To observe the claimed runtime, we give a high level description of a possible implementation. For any basis state $|\theta\rangle$, we may define a local unitary rotation, as in Example 1.3, for each bit of the representation, successively rotating the ancilla bit by some degrees each time. This does not violate the correctness, as the rotation map in the plane is linear. It is clear, that we only require d such gates and thus $\mathcal{O}(d)$ many local unitary gates. ■

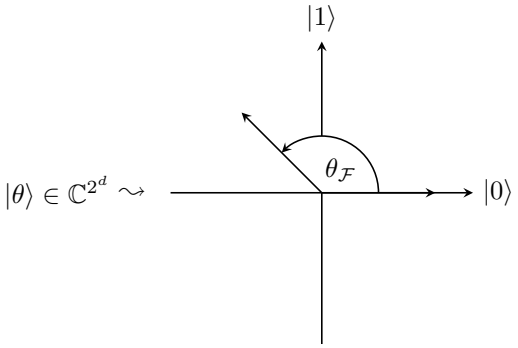


Figure 2: Unit vector rotations, controlled by qubit registers. Here for $\theta_{\mathcal{F}} = 3\pi/4$.

2.2 Quantum State Generation based on Efficiently Integrable Probability Distributions

Problem 2.4. (Quantum State Generation) Suppose one is given an initial state $|\psi\rangle \in \mathbb{C}^N$ and another state $\sum_{\tau=0}^{T-1} \alpha_\tau |\tau\rangle$. Give an efficient quantum algorithm, that performs the map $|\psi\rangle \mapsto \sum_{\tau=0}^{T-1} \alpha_\tau |\tau\rangle$.

This problem has been studied extensively. Approaches include the most direct method of successive rotation of the initial state into the target state [20, 21], which in both algorithms requires an exponential runtime. Aharonov et al. [22] have studied the problem in the framework of *Adiabatic Quantum*

Computation, a different framework for quantum algorithms using the so-called *Adiabatic theorem* from QM [1, p. 426 ff.]. There are no hardness results yet, to my knowledge. There is a simple combinatorial argument, that speaks against the existence of such circuits for any state, if we want to use a predefined set of finitely many gates [7, pp. 198-200]. One could say that no finite system of gates is complete wrt. efficient quantum state generation. Consider the following theorem with contained proof.

Theorem 2.5. Let $f, g \in \mathbb{N}_{\geq 1}$. Using g many efficient n-qubit-gates, each acting on at most f qubits, one can generate at most $\binom{n}{f}^g m \in \mathcal{O}(n^{fgm})$ states using m gates from $|0\rangle$.

We shall discuss a classical paper from 2002 by the researchers Grover and Rudolph [23]. It focuses on the case, where the coefficients $\alpha_\tau, \tau \in [0, T-1]_\mathbb{N}$ are given by *efficiently integrable probability density functions*.

Definition 2.6. Let $I \subseteq \mathbb{R}$, $I \neq \emptyset$, be compact and connected. We call a Riemann-integrable function $f\colon I \to \mathbb{R}$ *efficiently integrable*, if for any $x_0, x_1 \in I$ with $x_0 \leq x_1$, we can compute or at approximate $\int_{x_0}^{x_1} f$ in polynomial time.

The following theorem summarizes the result.

Theorem 2.7 (Quantum State Generation using Efficiently Integrable Probability Distributions)**.** Let $x_L^{m,i} \coloneqq i/2^m, x_R^i \coloneqq (i+1)/2^m$ for any $m \in \mathbb{N}_{\geq 1}, i \in [0, 2^m - 1]_\mathbb{N}$. For an arbitrary quantum state $\sum_{\tau=0}^{T-1} \alpha_\tau |\tau\rangle$ with $t \in \mathbb{N}_{\geq 1}, T \coloneqq 2^t$, s.t. there is a classically efficiently integrable probability density function $p\colon [0,1] \to [0,1]$ with $\alpha_\tau = \sqrt{\int_{[x_L^{t,\tau}, x_R^{t,\tau}]} p}$, there is a quantum algorithm, that solves the state generation problem $|0\rangle \mapsto \sum_{\tau=0}^{T-1} \alpha_\tau |\tau\rangle$ up to an arbitrary precision using some number of helper qubits, whilst being polynomial in t.

Proof. The squared amplitude magnitudes of our goal quantum state $\{|\alpha_\tau|^2 \mid \tau \in [0, T-1]_\mathbb{N}\}$ form a discrete probabiliy distribution. The probability space corresponds to the tuple $([0, T-1]_\mathbb{N}, \Pr)$ with $\Pr\colon [0, T-1]_\mathbb{N} \to [0,1], \tau \mapsto |\alpha_\tau|^2$. We perform t successive and even divisions of the interval $[0,1]$ and associate with each of these $t+1$ intervals a probability distribution $\{p_i^m \mid i \in [0, 2^m - 1]_\mathbb{N}\}$, where $m \in [0, t]_\mathbb{N}$ and:

$$p_i^m \coloneqq \sum_{\tau=2^{t-m}i}^{2^{t-m}(i+1)-1} p_\tau^t \tag{2.2.1}$$

Note that $p_0^0 = 1$, $p_\tau^t = |\alpha_\tau|^2$ by construction and especially:

$$\sum_{\tau=2^{t-m}i}^{2^{t-m}(i+1)-1} p_\tau^t = \sum_{\tau=2^{t-m}i}^{2^{t-m}(i+1)-1} \int_{x_L^{t,\tau}}^{x_R^{t,\tau}} p = \int_{x_L^{m,i}}^{x_R^{m,i}} p \tag{2.2.2}$$

Since p is classically efficiently computable, we can use our knowledge from quantum computability theory. We can construct a set of functions f_m as follows:

$$f_m\colon [0, 2^m - 1]_\mathbb{N} \to [0,1], i \mapsto \frac{\int_{x_L^{m,i}}^{\frac{x_L^{m,i} + x_R^{m,i}}{2}} p(x)\,dx}{\int_{x_L^{m,i}}^{x_R^{m,i}} p(x)\,dx} \overset{(1)}{=} \frac{\int_{x_L^{m+1,2i}}^{x_R^{m+1,2i}} p(x)\,dx}{\int_{x_L^{m,i}}^{x_R^{m,i}} p(x)\,dx} = \frac{p_{2i}^{m+1}}{p_i^m} \tag{2.2.3}$$

(1) By definition: $(x_L^{m,i} + x_R^{m,i})/2 = (i+i+1)/2^{m+1} = x_R^{m+1,2i}, x_L^{m,i} = 2i/2^{m+1} = x_L^{m+1,2i}$.

The idea is to extend some current m-qubit register, that is initialized with amplitudes from the target distribution, by one qubit each time. Wlog., we may assume $m \geq 1$ for the indices in the following calculations. If we have not initialized any qubit register yet, we can still apply the following analogously. Assume that we have already initialized this distribution-based quantum state for $m < t$ many qubits and are not finished, meaning that we have a register:

$$\sum_{i=0}^{2^m-1} \sqrt{\sum_{\tau=2^{t-m}i}^{2^{t-m}(i+1)-1} p_\tau}\, |i\rangle = \sum_{i=0}^{2^m-1} \sqrt{p_i^m}\, |i\rangle \tag{2.2.4}$$

Denote the unitary:

$$U_{f_m} : \mathbb{C}^{2^{m+d}} \to \mathbb{C}^{2^{m+d}}, |x\rangle |y\rangle \mapsto |x\rangle \left|y \oplus \arccos\left(\sqrt{f_m(x)}\right)\right\rangle =: |x\rangle |y \oplus \theta_x\rangle \tag{2.2.5}$$

Where $d \in \mathbb{N}_{\geq 1}$ is an arbitrary amount of auxiliary qubits and the exclusive disjunction is taken bitwise. Also note that the computed arccos and $\sqrt{...}$ functions are approximations of the corresponding real-valued functions. We leave this part to numerical mathematicians and add an additional qubit to this register and perform the computation:

$$\sum_{i=0}^{2^m-1} \sqrt{p_i^m} |i\rangle |0...0\rangle |0\rangle \xmapsto{U_{f_m} \times E_2} \sum_{i=0}^{2^m-1} \sqrt{p_i^m} |i\rangle |\theta_i\rangle |0\rangle \tag{2.2.6}$$

$$\xmapsto{E_{2^m} \times \mathrm{CR}_d} \sum_{i=0}^{2^m-1} \sqrt{p_i^m} |i\rangle |\theta_i\rangle (\cos(\theta_i) |0\rangle + \sin(\theta_i) |1\rangle) \tag{2.2.7}$$

$$\xmapsto{U_{f_m}^\dagger \times E_2} \sum_{i=0}^{2^m-1} \sqrt{p_i^m} |i\rangle |0\rangle (\cos(\theta_i) |0\rangle + \sin(\theta_i) |1\rangle) \tag{2.2.8}$$

$$\xmapsto{E_{2^{m+d-1}} \times \mathrm{SWAP}} \sum_{i=0}^{2^m-1} \sqrt{p_i^m} |i\rangle |0\rangle (\cos(\theta_i) |0\rangle + \sin(\theta_i) |1\rangle) |0\rangle \tag{2.2.9}$$

$$\overset{(1)}{\rightsquigarrow} \sum_{i=0}^{2^m-1} \sqrt{p_i^m} |i\rangle (\cos(\theta_i) |0\rangle + \sin(\theta_i) |1\rangle) |0\rangle \tag{2.2.10}$$

$$= \sum_{i=0}^{2^m-1} \sqrt{p_i^m} |i\rangle \left(\sqrt{f_m(i)} |0\rangle + \sqrt{1 - f_m(i)} |1\rangle\right) |0\rangle \tag{2.2.11}$$

(1) We perform the previous swap operation $d-1$ additional times to push out the remaining helper bits.

Since we uncomputed the helper register, it can be reused for other tasks. Although it is not obvious, the following computation shows that this corresponds to our target state.

$$\sum_{i=0}^{2^m-1} \sqrt{p_i^m} |i\rangle \left(\sqrt{f_m(i)} |0\rangle + \sqrt{1 - f_m(i)} |1\rangle\right) = \sum_{i=0}^{2^m-1} \sqrt{p_i^m} |i\rangle \left(\sqrt{\frac{p_{2i}^{m+1}}{p_i^m}} |0\rangle + \sqrt{\frac{p_i^m - p_{2i}^{m+1}}{p_i^m}} |1\rangle\right) \tag{2.2.12}$$

$$\overset{(1)}{=} \sum_{i=0}^{2^m-1} \sqrt{p_i^m} |i\rangle \left(\sqrt{\frac{p_{2i}^{m+1}}{p_i^m}} |0\rangle + \sqrt{\frac{p_{2i+1}^{m+1}}{p_i^m}} |1\rangle\right) \tag{2.2.13}$$

$$= \sum_{i=0}^{2^{m+1}-1} \sqrt{p_i^{m+1}} |i\rangle \tag{2.2.14}$$

(1) Just to be precise, we calculate this by definition. Some index play yields:

$$p_i^m - p_{2i}^{m+1} = \textstyle\sum_{\tau=2^{t-m}i}^{2^{t-m}(i+1)-1} p_\tau - \sum_{\tau=2^{t-m-1}2i}^{2^{t-m-1}(2i+1)-1} p_\tau = \sum_{\tau=2^{t-m}i}^{2^{t-m}i+2^{t-m}-1} p_\tau - \sum_{\tau=2^{t-m}i}^{2^{t-m}i+2^{t-m-1}-1} p_\tau \tag{2.2.15}$$

$$= \textstyle\sum_{\tau=2^{t-m}i+2^{t-m-1}}^{2^{t-m}i+2^{t-m}-1} p_\tau = \sum_{\tau=2^{t-m-1}(2i+1)}^{2^{t-m-1}(2i+2)-1} p_\tau = p_{2i+1}^{m+1} \tag{2.2.16}$$

Which was the desired state. So the trick here is to use the angle approximation as an angle in a rotation, but by that we introduce it as an amplitude, which gives the desired result.
All gates are efficiently implementable. Every iteration requires polynomial time. In total, we iterate t times, concluding the claimed runtime for $d \in \mathcal{O}(1)$. ■

Remark 2.8. One may notice

- one remarkable aspect of quantum computing also shows in this proof: Each construction iteration, the superposition of states gets doubled in $\mathcal{O}(1)$ quantum runtime. This choice of the clock register coefficients is also due to the error analysis.
- that the requirement of efficient integrability may be relaxed by the requirement of efficient integrability over the parts of the interval divisions considered.

We restrict ourselves to the interval $[0, 1]$ in both domain and image for simplicity and the procedure, but one can think about generalizing the result by adjusting the interval division and possibly introducing special indices for subintervals starting in $-\infty$ or ending in ∞.

Remark 2.9 (Discussion and Outlook). The proof of the above theorem from Grover and Rudolph shows, that, we can, in principle, construct any arbitrary quantum state, given the coefficients. A general, but inefficient integration can be achieved by the classical function just summing up the coefficients of the contained smallest intervals of level t in the current subinterval. Efficient schemes, such as utilizing binary trees, which may remind a computational geometrician of interval trees [24, p. 220-226], may allow for the logarithmic lookup of these sums, but such a data structure itself will still require $\mathcal{O}(N \log(N))$ space complexity. The idea of employing tree-like structures has been studied in a PhD thesis [12, pp. 23-27].

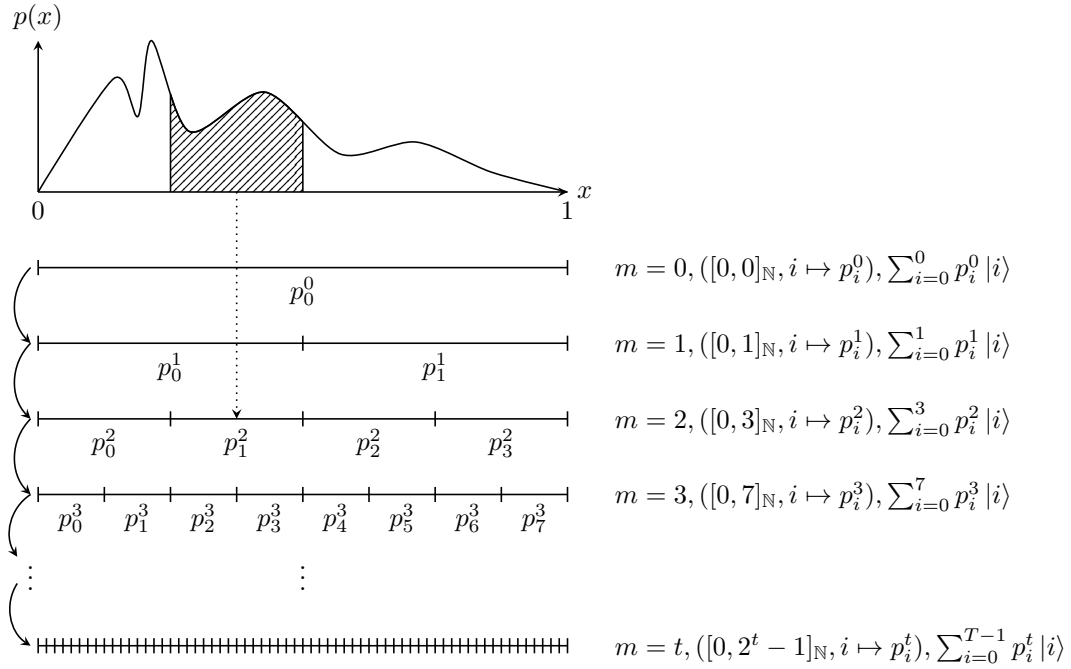


Figure 3: Sketch for understanding the divisions. Here for $t = 6$ and captions only in the first four divisions to avoid cluttering the sketch. The vertical axis has no markings as the image of the function p is drawn solely for illustration. On the right, the associated value of m, the associated discrete probability space and the corresponding state are denoted. The arrows on the left illustrate the direction of the inductive algorithm by Grover and Harris. The part of the area under the curve of p, which gives p_1^2, has been highlighted.

2.3 Quantum Mechanical Metrics

State Similarity

Let $|\varphi\rangle, |\psi\rangle \in \mathbb{C}^n$ be two quantum states for this part. To compare the similarity between them, one can use the introduced standard norm and define a metric by $\||\varphi\rangle - |\psi\rangle\|$.

Theorem 2.10. The tuple $(\{|\psi\rangle \in \mathbb{C}^n \mid \||\psi\rangle\| = 1\}, d_n)$ is a metric space, where d_n is the map

$$d_n \colon \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{R}_{\geq 0}, (|\varphi\rangle, |\psi\rangle) \mapsto \||\varphi\rangle - |\psi\rangle\| \tag{2.3.1}$$

The proof can be found in [2, p. 1-2], the definition of a metric can be found in [2, p. 551]. This theorem, to us, has semantical meaning. The complex vector norm induces a metric, meaning that we can measure the similarity between states by computing the above formula. A short rewrite yields the following theorem.

Theorem 2.11. It holds, that

$$\| |\varphi\rangle - |\psi\rangle \| = \sqrt{2(1 - \mathrm{Re}(\langle\varphi|\psi\rangle))} \tag{2.3.2}$$

Proof. We use the additivity of the complex standard product in both components and compute:

$$\begin{aligned} \| |\varphi\rangle - |\psi\rangle \|^2 &= \langle |\varphi\rangle - |\psi\rangle | |\varphi\rangle - |\psi\rangle \rangle && (2.3.3)\\ &= \langle\varphi|\varphi\rangle - \langle\psi|\varphi\rangle - \langle\varphi|\psi\rangle + \langle\psi|\psi\rangle && (2.3.4)\\ &\overset{(1)}{=} 2(1 - \mathrm{Re}(\langle\varphi|\psi\rangle)) && (2.3.5) \end{aligned}$$

(1) We use, that $\| |\varphi\rangle \|^2 = \| |\psi\rangle \|^2 = 1$. Furthermore, we have

$$\langle\psi|\varphi\rangle + \langle\varphi|\psi\rangle = \sum_{i=1}^{n} \varphi_i \psi_i^* + \sum_{i=1}^{n} \varphi_i^* \psi_i \tag{2.3.6}$$

However, $\varphi_i\psi_i^* + \varphi_i^*\psi_i = 2\varphi_{i1}\psi_{i1} + 2\varphi_{i2}\psi_{i2}$. So $\langle\psi|\varphi\rangle + \langle\varphi|\psi\rangle = 2\,\mathrm{Re}(\langle\varphi|\psi\rangle)$.

■

Operator Similarity

Similarly to state similarity, there are norms for operators, which allow us to, for instance, analyze the error of a quantum algorithm. With [2, p. 51] in mind, we introduce the following theorem.

Theorem 2.12. The following map is a norm, the so-called *operator norm*:

$$\|\cdot\| \colon \mathbb{C}^{n\times n} \to \mathbb{R}_{\geq 0}, A \mapsto \max_{\substack{|\varphi\rangle \in \mathbb{C}^n \\ \| |\varphi\rangle \| = 1}} \| A\,|\varphi\rangle \| \tag{2.3.7}$$

Proof. Let $S \coloneqq \{|\varphi\rangle \in \mathbb{C}^n \mid \| |\varphi\rangle \| = 1\}$. The map is well-defined, since linear maps over $\mathbb{C}^n$ are continuous and S is closed and bounded by definition wrt. the standard topology over $\mathbb{C}^n$, thus by Heine-Borel [8, p. 41] compact. So the maximum always exists [8, p. 43]. To be precise, the closedness is due to one being able to span a line between a point outside of S and 0 and taking the distance to the point hit on S as the associated open set for the point, the boundedness, due to the fact, that $\max\{\| |v\rangle \|\} = 1 < 2$, so $S \subset \{v \in \mathbb{C}^n \mid \|v\| < 2\}$.
The homogenity follows from $\| (zA)\,|\varphi\rangle \| = |z| \| A\,|\varphi\rangle \|$, which carries over into the maximum. The triangle inequality also carries over from the norm of $\mathbb{C}^n$. Lastly, the positive-definiteness is obtained by observing, that if the vectors of an arbitrary base are all mapped to zero, the linear map itself must be zero. This concludes the proof. ■

Example 2.13. Consider the operator given by

$$A \coloneqq \begin{pmatrix} 1 & 0 \\ i & 0 \end{pmatrix} \tag{2.3.8}$$

Then for any $(b_1, b_2) \in \mathbb{C}^2$, $\left\| A \begin{pmatrix} b_1 & b_2 \end{pmatrix}^t \right\|^2 = 2|b_1|^2$, so $\|A\| = \sqrt{2}$.

This norm also induces a metric, as the norm for states did.

Remark 2.14 (Comparing Operators). To compare two operators U and U' and prove a bound for the operator distance $\|U - U'\|$, it suffices to bound the distance $\|U\,|\varphi\rangle - U'\,|\varphi\rangle\|$, with $|\varphi\rangle$ being an arbitrary quantum state.

2.4 Qutrits

Whilst qubits are the quantum equivalent of bits, *qutrits* imitate the concept of a *trit*. Such electrical devices allow us to store three states, instead of two. In some systems, we denote them as $0, 1, 2$ and in others in form of the *balanced representation* $-1, 0, 1$ for convenience [25, p. 1]. A qutrit is a member of $\mathbb{C}^3$. We denote the canonical base as $\{|0\rangle, |1\rangle, |2\rangle\}$ [26, pp. 2-3].
We want to study three-dimensional rotations of qutrits. First, we give a lemma, which we derive from [10, pp. 70-75].

Definition 2.15. Let

$$\times \colon \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3, (u, v) \mapsto \begin{pmatrix} u_2v_3 - u_3v_2 \\ u_3v_1 - u_1v_3 \\ u_1v_2 - u_2v_1 \end{pmatrix} \tag{2.4.1}$$

be the so-called *cross product*.

Lemma 2.16. If $\{u, v\} \subseteq \mathbb{R}^3$ is an orthonormal system, then $\{u, v, u \times v\}$ is an orthonormal basis.

The proof can be found in the above-mentioned literature. When constructing three-dimensional unitaries, the cross product proves to be useful, as we will see in the proof of the following theorem.

Theorem 2.17 (Three-Dimensional Qutrit Rotation)**.** For a $m \in \mathbb{N}_{\geq 1}$-qubit-register with an appended qutrit and classically efficiently computable functions $f \colon \mathbb{R} \to \mathbb{R}$, $g \colon \mathbb{R} \to \mathbb{R}$, s.t. $f^2 + g^2 \leq 1$ and $g^2(x) \neq 1$ for all $x \in \mathbb{R}$, there is an efficient quantum algorithm, that achieves for any desirable precision

$$\mathbb{C}^{2^m \cdot 3} \to \mathbb{C}^{2^m \cdot 3}, |\lambda\rangle |0\rangle \mapsto |\lambda\rangle \left(\sqrt{1 - f^2(\lambda) - g^2(\lambda)} |0\rangle + f(\lambda) |1\rangle + g(\lambda) |2\rangle\right) \tag{2.4.2}$$

Proof. We start off with a geometric argument on three-dimensional rotations. One notices, that the desired state of the qutrit is normalized, and that the amplitudes of the state vector are real. We can imagine the problem as moving the vector $|0\rangle$ on the real unit sphere S^2 into our target state, as depicted in Figure 4.
We now directly explain, how to rotate the qutrit state $|0\rangle$ into an arbitrary qutrit state of form $|\xi\rangle \in \mathbb{R}^3$. Let $P_{01} \coloneqq |0\rangle\langle 0| + |1\rangle\langle 1|$. $\varphi \coloneqq \sphericalangle(|0\rangle, P_{01}|\xi\rangle), \psi \coloneqq \sphericalangle(P_{01}|\xi\rangle, |\xi\rangle)$. Let $(\varphi, \psi) = (0, \pi/2)$, if $|\xi\rangle_3 = 1$ and $(\varphi, \psi) = (0, 3\pi/2)$ for $|\xi\rangle_3 = -1$. Note, that P_{01} performs the projection onto the plane spanned by $\{|0\rangle, |1\rangle\}$. First, we rotate $|0\rangle$ along the plane spanned by $\{|0\rangle, |1\rangle\}$ by angle φ. Then, we rotate the resulting vector $P_{01}|\xi\rangle$ by the angle ψ along the plane spanned by $\{P_{01}|\xi\rangle, |2\rangle\}$ into $|\xi\rangle$.
The two-dimensional standard rotation matrix, as seen in Lemma 2.3, is for some $\theta \in (-\pi, \pi]$

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \tag{2.4.3}$$

In three dimensions, we can always fix one coordinate and look at the cartesian coordinate system, that is spanned, when the canonical unit vector of the fixed coordinate points upwards and the others point downwards. Especially, $|2\rangle$ is to the right of $|0\rangle$. Otherwise, $|0\rangle$ is right of $|1\rangle$ and $|1\rangle$ is right of $|2\rangle$. To rotate in these induced planes, we look at the rotation matrices of form

$$R_x(\theta) \coloneqq \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix}, R_y(\theta) \coloneqq \begin{pmatrix} -\sin(\theta) & 0 & \cos(\theta) \\ 0 & 1 & 0 \\ \cos(\theta) & 0 & \sin(\theta) \end{pmatrix}, R_z(\theta) \coloneqq \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.4.4}$$

Pay special note to $R_y(\theta)$. We apply $R_z(\varphi)$ to obtain

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \mapsto \begin{pmatrix} \cos(\varphi) \\ \sin(\varphi) \\ 0 \end{pmatrix} \tag{2.4.5}$$

and then

$$\begin{pmatrix} \cos(\varphi) \\ \sin(\varphi) \\ 0 \end{pmatrix} \mapsto \cos(\psi) \begin{pmatrix} \cos(\varphi) \\ \sin(\varphi) \\ 0 \end{pmatrix} + \sin(\psi) |2\rangle = \begin{pmatrix} \cos(\varphi)\cos(\psi) \\ \sin(\varphi)\cos(\psi) \\ \sin(\psi) \end{pmatrix} \tag{2.4.6}$$

To get geometrical intuition for this result, we can observe, that as we look at the latter described plane, the coefficients of this linear combination correspond to the lengths of the vectors for the final state. We want a unitary, which is able to rotate $|0\rangle$ as described. In other words, we are searching for values $\alpha, \beta, \gamma, \delta, \varepsilon, \zeta \in \mathbb{C}$, s.t. we have a unitary matrix of form

$$\begin{pmatrix} \cos(\varphi)\cos(\psi) & \alpha & \beta \\ \sin(\varphi)\cos(\psi) & \gamma & \delta \\ \sin(\psi) & \varepsilon & \zeta \end{pmatrix} \tag{2.4.7}$$

We can derive at least two such unitaries. First, observe, that the norm of the first column is 1, since Theorem B.3 gives us:

$$(\cos^2(\varphi) + \sin^2(\varphi))\cos^2(\psi) + \sin^2(\psi) = 1 \tag{2.4.8}$$

By inspecting the coefficients closely, we suggest the vector:

$$\begin{pmatrix} -\cos(\varphi)\sin(\psi) \\ -\sin(\varphi)\sin(\psi) \\ \cos(\psi) \end{pmatrix} \tag{2.4.9}$$

To form an orthonormal basis of a 2-dimensional subspace of $\mathbb{C}^3$. And indeed, we have for the inner product

$$\left\langle \begin{array}{c} \cos(\varphi)\cos(\psi) \\ \sin(\varphi)\cos(\psi) \\ \sin(\psi) \end{array} \middle| \begin{array}{c} -\cos(\varphi)\sin(\psi) \\ -\sin(\varphi)\sin(\psi) \\ \cos(\psi) \end{array} \right\rangle = -(\cos^2(\varphi) + \sin^2(\varphi))\cos(\psi)\sin(\psi) + \cos(\psi)\sin(\psi) = 0 \tag{2.4.10}$$

The vector is also normalized, due to

$$(\cos^2(\varphi) + \sin^2(\varphi))\sin^2(\psi) + \cos^2(\psi) = 1 \tag{2.4.11}$$

In the above vector, we negated the first two components. One can also negate only the third one, but for $(\varphi, \psi) = (0, 0)$, that would give us $-|2\rangle$. We choose the first version, due to preference. Furthermore, to construct the third vector, we can take cross product, see previously Lemma 2.16, of both vectors to obtain a vector, that is orthogonal to both and even normalized. We have

$$\begin{pmatrix} \cos(\varphi)\cos(\psi) \\ \sin(\varphi)\cos(\psi) \\ \sin(\psi) \end{pmatrix} \times \begin{pmatrix} -\cos(\varphi)\sin(\psi) \\ -\sin(\varphi)\sin(\psi) \\ \cos(\psi) \end{pmatrix} \tag{2.4.12}$$

$$= \begin{pmatrix} \sin(\varphi)\cos^2(\psi) + \sin^2(\psi)\sin(\varphi) \\ -\sin^2(\psi)\cos(\varphi) - \cos(\varphi)\cos^2(\psi) \\ -\cos(\varphi)\cos(\psi)\sin(\varphi)\sin(\psi) + \sin(\varphi)\cos(\psi)\cos(\varphi)\sin(\psi) \end{pmatrix} = \begin{pmatrix} \sin(\varphi) \\ -\cos(\varphi) \\ 0 \end{pmatrix} \tag{2.4.13}$$

The obtained vector is normalized and, with the other two vectors, forms an orthogonal base of $\mathbb{C}^3$ with the other two vectors. We get the matrix

$$R(\varphi, \psi) \coloneqq \begin{pmatrix} \cos(\varphi)\cos(\psi) & -\cos(\varphi)\sin(\psi) & \sin(\varphi) \\ \sin(\varphi)\cos(\psi) & -\sin(\varphi)\sin(\psi) & -\cos(\varphi) \\ \sin(\psi) & \cos(\varphi) & 0 \end{pmatrix} \tag{2.4.14}$$

This matrix is unitary, according to our derivation and Theorem 1.4. We turn to the original problem. Using the first column of the matrix, we get the desired condition:

$$|0\rangle \mapsto \begin{pmatrix} \cos(\varphi)\cos(\psi) \\ \sin(\varphi)\cos(\psi) \\ \sin(\psi) \end{pmatrix} = \begin{pmatrix} \sqrt{1 - f^2(\lambda) - g^2(\lambda)} \\ f(\lambda) \\ g(\lambda) \end{pmatrix} \tag{2.4.15}$$

From which we get $\psi \coloneqq \arcsin(g(\lambda))$ and $\varphi \coloneqq \arcsin\left(\frac{f(\lambda)}{\sqrt{1-g^2(\lambda)}}\right)$.

Append $n, o \in \mathbb{N}_{\geq 1}$ zeroed-out qubits to the first register and perform the map

$$|\lambda\rangle |0\rangle |0\rangle \mapsto |\lambda\rangle \left| \left(\arcsin\left(\frac{f(\lambda)}{\sqrt{1-g^2(\lambda)}} \right), \arcsin(g(\lambda)) \right) \right\rangle \tag{2.4.16}$$

Here, we have stored the approximations of φ and ψ in the two auxiliary registers. It is also clear, that we can denote such a tuple in binary via an arbitrary encoding format. We define a unitary of form

$$\sum_{\theta=0}^{2^o-1} |\theta\rangle \langle\theta| \otimes R(0, \mathcal{F}(\theta)) + \sum_{\theta=2^o}^{2^{n+o}-1} |\theta\rangle \langle\theta| \otimes R(\mathcal{F}(\theta - 2^o), \mathcal{F}(\theta \bmod 2^o)) \tag{2.4.17}$$

similar to the proof of Lemma 2.3. This gives us the desired behavior of

$$|\lambda\rangle \left| \left(\arcsin\left(\frac{f(\lambda)}{\sqrt{1-g^2(\lambda)}} \right), \arcsin(g(\lambda)) \right) \right\rangle |0\rangle \tag{2.4.18}$$

$$\mapsto |\lambda\rangle |0\rangle |0\rangle \left(\sqrt{1 - f^2(\lambda) - g^2(\lambda)} |0\rangle + f(\lambda) |1\rangle + g(\lambda) |2\rangle \right) \tag{2.4.19}$$

by uncomputing the helper bits afterwards. It is clear, that the above instructions are all efficiently implementable. Thus, we are finished. ■

Remark 2.18 (On the Orientation of the Coordinate System). We have transformed cartesian system, spanned by $\{|0\rangle, |1\rangle, |2\rangle\}$, into another one, spanned by $\{|0\rangle, |2\rangle, -|1\rangle\}$, with the associated mappings of the canonical basis vectors in order of this enumeration. This may help visualize the transformation much better. Note that, furthermore, the vector product respects the *right hand rule* [10, pp. 70-75], related to Lenz's law from electrophysics [27, pp. 314-315]. By reversing the order in the vector product, we could obtain a map into the coordinate system, spanned by the vectors $\{|0\rangle, |2\rangle, |1\rangle\}$, in this order, but we refrain from doing that.

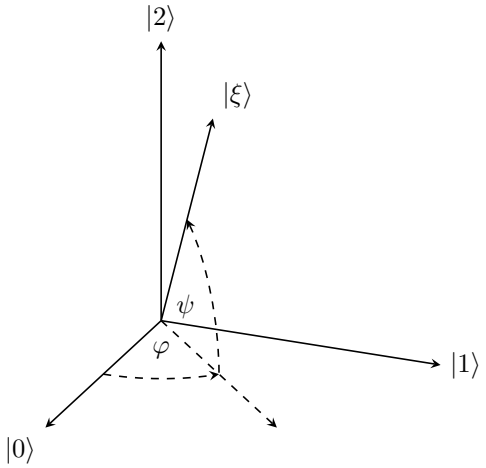


Figure 4: Rotating the qutrit state $|0\rangle$ according to some angles $\varphi, \psi \in (-\pi, \pi]$ into some state $|\xi\rangle \in \mathbb{R}^3 \subset \mathbb{C}^3$. Here illustrated for $|\xi\rangle \coloneqq \frac{1}{\sqrt{10}}(|0\rangle + |1\rangle) + \frac{2}{\sqrt{5}} |2\rangle$, thus $\varphi = \frac{\pi}{4}$ and $\psi = \arcsin\left(\frac{2}{\sqrt{5}}\right)$.

2.5 Amplitude Amplification

In 2000, Brassard et al. [28, pp. 4-10] studied the problem of boosting the success probability of an arbitrary quantum algorithm, i.e., the probability of a measurement yielding a desired result. Inspired by the ideas from Grover et al., the researchers developed the so-called *Amplitude Amplification* algorithm, to which this subsection is dedicated.

Grovers Multi-Search Algorithm

Grovers algorithm for a multi-search problem, as in [19, pp. 140-155], may be formulated the following way.

Theorem 2.19 (Grovers Algorithm for a Multi-Search Problem)**.** Given a function of form $f\colon [0, N-1]_{\mathbb{N}} \to \{0,1\}$, $N \coloneqq 2^n$ with $0 < |f^{-1}(1)| =: M < N$, there exists a quantum algorithm, that can find an element from $f^{-1}(1)$ in time $\mathcal{O}(\sqrt{N/M})$.

Remark 2.20 (Form of f)*.* If the cardinality of the domain of f is not a power of two, then f can be naturally extended to some function $\hat{f}$ by extending the count to the next power of two and mapping all of the new domain values to zero.

We want to now recall the idea of the multi-search version, as the version for single-search problems is naturally implied and works analogously. We shall further omit the analysis, as we will prove the more general case with AA. The structure of Grovers algorithm follows the following iteration: We first construct the uniform superposition state $H^{\otimes n}|0\rangle$, initialize a helper qubit to $(H \circ \mathrm{NOT})|0\rangle$ and then repeatedly apply the so-called *Grover operator* [19, p. 146]

$$G \coloneqq -H^{\otimes n} R_N H^{\otimes n} V_f \tag{2.5.1}$$

exactly

$$G(N,M) \approx \frac{\pi}{4} / \arcsin\left(\sqrt{\frac{M}{N}}\right) - \frac{1}{2} \tag{2.5.2}$$

times [19, p. 153-155], where $G(N,M) \in \mathbb{N}$ denotes the required number of iterations. As $G(N,M) \in \mathcal{O}(\sqrt{N/M})$ due to Lemma 3.11, this gives the claimed runtime. In the analysis of Grovers algorithm, V_f is interpreted as the operator mirroring the amplitudes to be boosted wrt. 0, whilst $-H^{\otimes n} R_N H^{\otimes n}$ mirrors all amplitudes wrt. the arithmetic mean of all amplitudes. This directly gives the geometric interpretation for the algorithm, that Grovers procedure successively rotates the uniform superposition state into the boosted state. The operators $R_N, V_f \in \mathbb{C}^{N\times N}$ here are defined via the following actions on any canonical basis vector $|x\rangle$ of $\mathbb{C}^N$:

$$R_N|x\rangle \coloneqq \begin{cases} -|x\rangle & x = 0 \\ |x\rangle & x \neq 0 \end{cases} \qquad V_f|x\rangle \coloneqq \begin{cases} -|x\rangle & x \in f^{-1}(1) \\ |x\rangle & x \notin f^{-1}(0) \end{cases} \tag{2.5.3}$$

Let it be noted, that the required helper qubit is used by V_f and omitted here for simplicity. Denoting $V_f \in \mathbb{C}^{N\times N}$ is thus technically wrong, but we do not loose any generality. Both gates can be efficiently implemented [19, pp. 144-145].

The General Case

Now to the more general case in AA. Consider a measurement-free (except for helper qubits) quantum algorithm acting on n qubits $U \in \mathbb{C}^{N\times N}$ and a Boolean function $\chi\colon \{0,1\}^n \to \{0,1\}$. Suppose we wish to measure the state $|\Psi\rangle \coloneqq U|0\rangle$ wrt. the observable $\{\mathrm{span}(\{|k\rangle \mid k \in \chi^{-1}(0)\}), \mathrm{span}(\{|k\rangle \mid k \in \chi^{-1}(1)\})\}$ to obtain the index of the subspace given by $\chi^{-1}(1)$ with probability $p \coloneqq \langle P_1|\Psi\rangle | P_1|\Psi\rangle\rangle \in (0,1)$ wlog., where $P_i \coloneqq \sum_{x\in\chi^{-1}(i)} |x\rangle\langle x|$ for $i \in \mathbb{F}_2$. We especially want to boost that probability of success. The following formulation is directly based off the paper by Brassard et al..

Denote $|\Psi_i\rangle \coloneqq P_i|\Psi\rangle$ as well for $i \in \mathbb{F}_2$. Initialize a n-qubit register to $U|0\rangle$, and an ancilla qubit to $(H \circ \mathrm{NOT})|0\rangle$, but we omit it as explained above. We now define an operator $Q \in \mathbb{C}^{N\times N}$ via

$$Q \coloneqq -U R_N U^\dagger V_\chi \tag{2.5.4}$$

where V_χ corresponds to V_f via $f \coloneqq \chi$. This operator corresponds to a direct generalization of the Grover operator from Equation (2.5.1).

Lemma 2.21. Let $\theta_p \coloneqq \arcsin\left(\sqrt{p}\right)$. After $j \in \mathbb{N}$ iterations of Q on $|\Psi\rangle$, we have

$$Q^j |\Psi\rangle = \frac{1}{\sqrt{p}} \sin((2j+1)\theta_p) |\Psi_1\rangle + \frac{1}{\sqrt{1-p}} \cos((2j+1)\theta_p) |\Psi_0\rangle \tag{2.5.5}$$

This result is presented in [28, pp. 5-7], we give the argument in more detail.

Proof. The proof is divided into the following parts:

(i) We first calculate the states $Q|\Psi_1\rangle$ and $Q|\Psi_0\rangle$, which will give us, that the operator only acts in the two-dimensional subspace $\operatorname{span}(\{|\Psi_1\rangle, |\Psi_0\rangle\})$.

(ii) After rewriting Q wrt. the subspace spanned by the images from Section 2.5, we calculate its eigenvalues and thus span an eigenbasis for the mentioned subspace.

(iii) The initial state $|\Psi\rangle$ is rewritten in the subspace spanned by the eigenvectors, from which we can calculate $Q^j |\Psi\rangle$.

(i) The behavior of Q on the states $|\Psi_0\rangle$ and $|\Psi_1\rangle$ is given by:

$$Q|\Psi_1\rangle \overset{(1)}{=} U(E_N - 2|0\rangle\langle 0|)U^\dagger |\Psi_1\rangle \overset{(2)}{=} |\Psi_1\rangle - 2|\Psi\rangle\langle\Psi|\Psi_1\rangle \overset{(3)}{=} (1-2p)|\Psi_1\rangle - 2p|\Psi_0\rangle \tag{2.5.6}$$

$$Q|\Psi_0\rangle = -U(E_N - 2|0\rangle\langle 0|)U^\dagger |\Psi_0\rangle = -|\Psi_0\rangle + 2|\Psi\rangle\langle\Psi|\Psi_0\rangle \overset{(4)}{=} 2(1-p)|\Psi_1\rangle + (1-2p)|\Psi_0\rangle \tag{2.5.7}$$

(1) Consider the definitions of V_χ and R_N as in Equation (2.5.3), from which we directly have $R_N = E_N - 2|0\rangle\langle 0|$.

(2) Using $UE_NU^\dagger = E_N$ and $2U|0\rangle\langle 0|U^\dagger = 2U|0\rangle(U|0\rangle)^\dagger = 2|\Psi\rangle\langle\Psi|$.

(3) By $|\Psi\rangle = |\Psi_1\rangle + |\Psi_0\rangle$ and the orthogonality relations.

(4) As $\langle\Psi|\Psi_0\rangle = \langle\Psi|(|\Psi\rangle - |\Psi_1\rangle) = 1-p$.

We rewrite Q wrt. the space $H_\Psi \coloneqq \operatorname{span}(\{|\Psi_1\rangle, |\Psi_0\rangle\})$ as

$$Q_\Psi \coloneqq \begin{pmatrix} 1-2p & 2(1-p) \\ -2p & 1-2p \end{pmatrix} \tag{2.5.8}$$

Since $|\Psi\rangle \in H_\Psi$, we may repeatedly apply Q and remain in H_Ψ, as in Grovers algorithm.

(ii) We calculate the eigenvalues and eigenvectors of Q_Ψ. The characteristic polynomial gives via Sarrus' rule

$$\det(Q_\Psi - \lambda_\pm E_2) = (1-2p-\lambda_\pm)^2 + 4p(1-p) = \lambda_\pm^2 - 2(1-2p)\lambda_\pm + 1 \tag{2.5.9}$$

and by that

$$\lambda_\pm = (1-2p) \pm \sqrt{(1-2p)^2 - 1} = 1 \pm i2\sqrt{p}\sqrt{1-p} - 2p \tag{2.5.10}$$

$$\overset{(1)}{=} \cos^2(\theta_p) + \sin^2(\theta_p) \pm i2\sin(\theta_p)\cos(\theta_p) - 2\sin^2(\theta_p) \tag{2.5.11}$$

$$= (\cos(\theta_p) \pm i\sin(\theta_p))^2 = e^{\pm i2\theta_p} \tag{2.5.12}$$

(1) Define $\theta_p \coloneqq \arcsin\left(\sqrt{p}\right) \in [0, \pi/2]$ and use Theorem B.3.

To obtain an eigenvector base for H_Ψ, consider the SLE $(Q_\Psi - \lambda_\pm E_2)|\Psi_\pm\rangle = 0$ for some $|\Psi_\pm\rangle \coloneqq \alpha_\pm|\Psi_1\rangle + \beta_\pm|\Psi_0\rangle$ with $\alpha_\pm, \beta_\pm \in \mathbb{C}$. We handle both λ_+ and λ_- in one calculation. Considering, that

$$1 - 2p - \lambda_\pm = 1 - 2p - (\sqrt{1-p} \pm i\sqrt{p})^2 = \mp i2\sqrt{p}\sqrt{1-p} \tag{2.5.13}$$

and dividing the SLEs first two rows by $2\sqrt{1-p}$ and $2\sqrt{p}$ respectively, we rewrite the SLE as

$$\begin{pmatrix} \mp i\sqrt{p} & \sqrt{1-p} \\ -\sqrt{p} & \mp i\sqrt{1-p} \end{pmatrix} \begin{pmatrix} \alpha_\pm \\ \beta_\pm \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{2.5.14}$$

Wlog. setting $\alpha_\pm \coloneqq 1/\sqrt{2p}$, we obtain

$$\begin{pmatrix} \mp \frac{i}{\sqrt{2}} + \sqrt{1-p}\beta_\pm \\ -\frac{1}{\sqrt{2}} \mp i\sqrt{1-p}\beta_\pm \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{2.5.15}$$

and thus $\beta_\pm = \pm\frac{i}{\sqrt{2}}\frac{1}{\sqrt{1-p}}$. An eigenbasis of H_Ψ is thus composed of the eigenvectors

$$|\Psi_\pm\rangle \coloneqq \alpha_\pm |\Psi_1\rangle + \beta_\pm |\Psi_0\rangle = \frac{1}{\sqrt{2}} \left(\frac{1}{\sqrt{p}} |\Psi_1\rangle \pm \frac{i}{\sqrt{1-p}} |\Psi_0\rangle \right) \tag{2.5.16}$$

(iii) Rewriting $|\Psi\rangle$ in these eigenbasis vectors gives using the square root of the full form of $\lambda_\pm$ as in Equation (2.5.13), i.e. $\sqrt{\lambda_\pm} = \sqrt{1-p} \pm i\sqrt{p} = e^{i\theta_p}$, gives

$$|\Psi\rangle = |\Psi_1\rangle + |\Psi_0\rangle = \sqrt{\frac{p}{2}}(|\Psi_+\rangle + |\Psi_-\rangle) + \frac{1}{i}\sqrt{\frac{1-p}{2}}(|\Psi_+\rangle - |\Psi_-\rangle) = \frac{-i}{\sqrt{2}}(e^{i\theta_p}|\Psi_+\rangle - e^{-i\theta_p}|\Psi_-\rangle) \tag{2.5.17}$$

As $|\Psi_+\rangle$ and $|\Psi_-\rangle$ are eigenvectors of Q_Ψ, we thus have

$$Q_\Psi^j |\Psi\rangle = \frac{-i}{\sqrt{2}} \left(\frac{1}{\sqrt{2p}}(e^{i(2j+1)\theta_p} - e^{-i(2j+1)\theta_p}) |\Psi_1\rangle + \frac{i}{\sqrt{2(1-p)}}(e^{i(2j+1)\theta_p} + e^{-i(2j+1)\theta_p}) |\Psi_0\rangle \right) \tag{2.5.18}$$

$$= \frac{-i}{2\sqrt{p}}(2i)\sin((2j+1)\theta_p)|\Psi_1\rangle + \frac{1}{2\sqrt{1-p}}(2)\cos((2j+1)\theta_p)|\Psi_0\rangle \tag{2.5.19}$$

$$= \frac{1}{\sqrt{p}}\sin((2j+1)\theta_p)|\Psi_1\rangle + \frac{1}{\sqrt{1-p}}\cos((2j+1)\theta_p)|\Psi_0\rangle \tag{2.5.20}$$

This concludes the proof. ■

Assuming, that we know the success probability p, we can directly use Lemma 2.21 to determine the number of iterations needed for producing a state close to $|\Psi_1\rangle$. The condition

$$\sin^2((2j+1)\theta_p) \approx 1 \tag{2.5.21}$$

for the success probablity of the state $Q^j|\Psi\rangle$ can be optimized by letting approximately $j \coloneqq \lfloor \pi/(4\theta_p) \rfloor$. As $\theta_p = \arcsin(\sqrt{p}) > \sqrt{p}$ following Lemma 3.11, we thus have a runtime of $\mathcal{O}(1/\sqrt{p})$.[5]

We now want to consider the case in which p is unknown [28, pp. 8-10]. The idea by Brassard et al. is to uniformly choose a number of applications of Q, which is exponentially bounded. To be precise, consider the following algorithm.

[5] Even $\Theta(1/\sqrt{p})$ for this general case due to lower-bound results for Grovers Algorithm, in which we will not dive into in this thesis.

Algorithm 1 Amplitude Amplification

Given: A unitary $U \in \mathbb{C}^{N\times N}$, with $N \coloneqq 2^n$, $n \in \mathbb{N}_{\geq 1}$, as well as a function $\chi\colon [0, N-1]_{\mathbb{N}} \to \{0,1\}$ with $\chi^{-1}(1) \notin \{\emptyset, [0, N-1]_{\mathbb{N}}\}$.

Return: A quantum state $|\Psi\rangle \in \mathbb{C}^N$, where the measurement of $U|0\rangle$ wrt. the observable $\{\mathrm{span}(\mathcal{B}_0), \mathrm{span}(\mathcal{B}_1)\}$ with $\mathcal{B}_i \coloneqq \{|x\rangle \mid x \in \chi^{-1}(i)\}$ for $i \in \mathbb{F}_2$ yielded a 1.

1: Let $l \coloneqq 0$, $M \coloneqq 0$ and let $c \in (1,2)$ be arbitrary, but fixed.
2: **while** true **do**
3: $\quad l \leftarrow l+1$, $M \leftarrow \lceil c^l \rceil$
4: $\quad$ Initialize $|\Psi\rangle \coloneqq U|0\rangle \in \mathbb{C}^N$, while considering the necessary ancilla qubit.
5: $\quad$ Measure $|\Psi\rangle$ wrt. $\{\mathrm{span}(\mathcal{B}_0), \mathrm{span}(\mathcal{B}_1)\}$, obtaining an index $z \in \mathbb{F}_2$.
6: $\quad$ **if** $z = 1$ **then**
7: $\quad\quad$ **return** $|\Psi\rangle$
8: $\quad$ **else**
9: $\quad\quad$ Initialize $|\Psi'\rangle \coloneqq U|0\rangle \in \mathbb{C}^N$, while considering the necessary ancilla qubit.
10: $\quad\quad$ Pick some $j \in [1, M]_{\mathbb{N}}$ uniformly at random.
11: $\quad\quad$ $|\Psi'\rangle \leftarrow Q^j |\Psi'\rangle$
12: $\quad\quad$ Measure $|\Psi'\rangle$ wrt. $\{\mathrm{span}(\mathcal{B}_0), \mathrm{span}(\mathcal{B}_1)\}$, obtaining an index $z' \in \mathbb{F}_2$.
13: $\quad\quad$ **if** $z' = 1$ **then**
14: $\quad\quad\quad$ **return** $|\Psi'\rangle$
15: $\quad\quad$ **else**
16: $\quad\quad\quad$ Go to step 3.

For the analysis, we need the following lemma.

Lemma 2.22. For any $\alpha \in \mathbb{C}$ and $m \in \mathbb{N}_{\geq 1}$, we have

$$\sum_{j=0}^{m-1} \cos((2j+1)\alpha) = \frac{\sin(2m\alpha)}{2\sin(\alpha)} \tag{2.5.22}$$

This lemma is taken from lemma 1 of a previous work on tight bounds for Grovers algorithm by Brassard and Boyer et al. [29, p. 3]. A similar analysis to the one presented in [28, pp. 9-10] is given there. We prove the lemma in the appendix, see Appendix A.

Lemma 2.23. Algorithm 1 runs in time $\mathcal{O}\left(\frac{1}{\sqrt{p}}\right)$.

Proof. We use the notation as in Lemma 2.21 and Algorithm 1. Analyzing the number of calls directly turns out to be quite difficult due to the Laplacian experiment in step Line 10. Brassard et al. thus suggest the following proof strategy: The expected probability of success in Line 12 is first lower-bounded and then the variable l is split at a point $M_0 \in \mathbb{N}$, chosen in dependence of θ_p and c. We then argue that both until M_0 is reached and afterwards, we require $\mathcal{O}\left(\frac{1}{\sqrt{p}}\right)$ applications of Q.

First, consider the case, where $p \geq 1/4$. Then the expected number of calls to Line 5 is

$$\sum_{n=0}^{\infty} (1-p)^n \geq \sum_{n=0}^{\infty} \frac{3}{4^n} = 4 \tag{2.5.23}$$

due to Theorem B.5.

Assume $p < 1/4$. Let $l \in \mathbb{N}_{\geq 1}$ be fixed and $M \coloneqq \lceil c^l \rceil$. Let $X\colon [1,M]_{\mathbb{N}} \to [0,1], j \mapsto \frac{1}{p}\sin^2((2j+1)\theta_p)$ denote the Laplacian random variable assigning each j the success probability of Line 12 as in Lemma 2.21. Then

$$\mathrm{E}[X] = \sum_{j=1}^{M} \frac{1}{M}\frac{1}{p}\sin^2((2j+1)\theta_p) \overset{(1)}{=} \frac{1}{2p}\left(1 - \frac{1}{M}\sum_{j=1}^{M}\cos((2j+1)(2\theta_p))\right) \tag{2.5.24}$$

$$\overset{(2)}{=} \frac{1}{2p}\left(1 - \frac{1}{2M}\frac{\sin(4(M+1)\theta_p)}{\sin(2\theta_p)} + \frac{1}{M}\cos(2\theta_p)\right) \tag{2.5.25}$$

$$\overset{(3)}{>} \frac{1}{2}\left(1 - \frac{1}{2M}\frac{\sin(4(M+1)\theta_p)}{\sin(\theta_p)}\right) \overset{(4)}{=} \frac{1}{2}\left(1 - \frac{1}{2M\sqrt{p}}\right) \tag{2.5.26}$$

(1) Using Lemma 3.3.

(2) Using Lemma 2.22.

(3) Consider the strict monotonicity of $\sqrt{\cdot}$ and arcsin in $(0,1]$. So $\theta_p \in (0, \arcsin(1/2))$. Use $\sin(2x) > \sin(x)$ for $x \in (0, \pi/4] \supset (0, \arcsin(1/2))$, as the sine is strictly monotonous in $[0, \pi/2]$. Use $p \in (0,1)$ as well. Also use $\cos|_{[0,\pi/2]} > 0$.

(4) By the definition of θ_p, see Lemma 2.21, we have $\sin(\theta_p) = \sqrt{p}$.

We shall from now on continue with the expected probability for success in Line 12. Whether the resulting lower bound is an actual probability solely depends on the value of l, so we may derive the condition

$$\frac{1}{2}\left(1 - \frac{1}{2M\sqrt{p}}\right) \in [0,1] \rightsquigarrow \frac{1}{2\sqrt{p}} \leq c^l \leq M^l \rightsquigarrow l \geq \log_c\left(\frac{1}{2\sqrt{p}}\right) \tag{2.5.27}$$

This observation leads to the following approach for the analysis: Until the condition in Equation (2.5.27) holds, we may iterate $\lceil \log_c(1/(2\sqrt{p})) \rceil$ times, requiring at most

$$\sum_{l=1}^{\left\lceil \log_c\left(\frac{1}{2\sqrt{p}}\right)\right\rceil} \lceil c^l \rceil \leq \sum_{l=0}^{\left\lceil \log_c\left(\frac{1}{2\sqrt{p}}\right)\right\rceil} (c^l + 1) = \frac{1 - c^{\left\lceil \log_c\left(\frac{1}{2\sqrt{p}}\right)\right\rceil + 1}}{1-c} + \left\lceil \log_c\left(\frac{1}{2\sqrt{p}}\right)\right\rceil \tag{2.5.28}$$

$$\leq \frac{1 - \frac{c^2}{2\sqrt{p}}}{1-c} + \left\lceil \log_c\left(\frac{1}{2\sqrt{p}}\right)\right\rceil \in \mathcal{O}\left(\frac{1}{\sqrt{p}}\right) \tag{2.5.29}$$

calls to Q using Theorem B.5. Let $M_0 \coloneqq c^{\left\lceil \log_c\left(\frac{1}{2\sqrt{p}}\right)\right\rceil}$. We have analyzed the asymptotic number of calls to Q until M has reached or surpassed M_0. After that, $c^l = M_0 c^i$ with $i \coloneqq l - \left\lceil \log_c\left(\frac{1}{2\sqrt{p}}\right)\right\rceil \geq 1$. By bounding $M_0 > \frac{1}{2\sqrt{p}}$ and $M > M_0 c^i$, we have a lower bound for the success probability of Line 12 by

$$\mathrm{E}[X] > \frac{1}{2}\left(1 - \frac{1}{2M_0 c^i \sqrt{p}}\right) > \frac{1}{2}\left(1 - \frac{1}{c^i}\right) \tag{2.5.30}$$

The complementary event, by the previous argument, is bounded from above by $1 - \mathrm{E}[X] \leq \frac{1}{2}\left(1 + \frac{1}{c^i}\right)$. Consider the condition $\frac{c}{2}\left(1 + \frac{1}{c^i}\right) < 1$, which gives $i > \log_c\left(\frac{c}{2-c}\right)$. Now looking at the expectation value of the geometric random variable counting the calls of Q after $M > M_0$, which is roughly $\sum_{i=1}^{\infty} M_0 c^i \left(\frac{1}{2}\left(1 + \frac{1}{c^i}\right)\right)^i$, we may conclude the statement, as after $\mathcal{O}(1)$ many steps involving $\mathcal{O}(M_0)$ applications of Q each, $\frac{c}{2}\left(1 + \frac{1}{c^i}\right) < 1$, where we can bound the limit of the associated geometric series by $\mathcal{O}(M_0)$. So we require a total of $\mathcal{O}(M_0) = \mathcal{O}(1/\sqrt{p})$ calls to Q, concluding the argument. ■

Theorem 2.24 (Amplitude Amplification)**.** Given a measurement-free quantum algorithm $U \in \mathbb{C}^{N \times N}$, $N \coloneqq 2^n$, $n \in \mathbb{N}$, which succeeds, i.e., gives a normalized projection into a subspace spanned by desirable basis vectors, with a possibly a priori unknown probability $p \in (0,1)$, we can perform a successful measurement using a quantum algorithm, that requires a runtime of $\mathcal{O}(1/\sqrt{p})$.

Corollary 2.25. Theorem 2.19 follows, especially for the case, when M is unknown. Consider $\chi \coloneqq f$ and the unitary $U \coloneqq H^{\otimes n}$ and the quantum algorithm, that initializes $U|0\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle$. Then measuring wrt. the observable $\{\mathrm{span}(\mathcal{B}_0), \mathrm{span}(\mathcal{B}_1)\}$ with the notation as in Algorithm 1 succeeds with probability M/N. Finding an element from $f^{-1}(1)$ thus requires $\mathcal{O}(\sqrt{N/M})$ applications of Q using AA. Note, that after the measurement, we need to measure again wrt. the observable $\{\mathrm{span}(|x\rangle) \mid x \in \{0,1\}^n\}$, as the state is then in $\mathrm{span}(\mathcal{B}_1)$.

Corollary 2.26. AA is analogously applicable in the case, where qutrits are used inside of the algorithm of concern, as we only have to adjust the operator Q and consider in the analysis, that the Boolean function maps from the qubit-qutrit canonical basis vector indices into $\{0,1\}$.

2.6 Quantum Phase Estimation

Problem 2.27. Let $U \in \mathbb{C}^{N\times N}$, $N \coloneqq 2^n$, $n \in \mathbb{N}_{\geq 1}$, be a unitary matrix with an eigenstate $|b\rangle \in \mathbb{C}^N$ of phase $\theta \in [0,1)$. The problem of calculating or approximating θ is called *quantum phase estimation* (QPE).

There is a general algorithm to this problem. The following theorem summarizes this classical result, which can be found in [7, pp. 221-226]. Figure 6 shows the circuit diagram of the general QPE algorithm.

Theorem 2.28 (General Quantum Phase Estimation)**.** Let $\varepsilon \in (0,1)$, $n \in \mathbb{N}_{\geq 1}$, $N \coloneqq 2^n$ and

$$t \geq n + \left\lceil \log\left(2 + \frac{1}{2\varepsilon}\right)\right\rceil \tag{2.6.1}$$

Given a unitary $U \in \mathbb{C}^{N\times N}$, an eigenstate $|b\rangle \in \mathbb{C}^N$ of U, an oracle for calculating the controlled unitaries $\hat{U}^{(2^k)} \colon \mathbb{C}^{2\cdot N} \to \mathbb{C}^{2\cdot N}$ achieving $|0\rangle|b\rangle \mapsto (|0\rangle + e^{2\pi i(2^k\theta)}|1\rangle)|b\rangle$ for arbitrary $k \in [0, 2^{t-1}]_{\mathbb{N}}$ and a unitary gate $\mathcal{B} \in \mathbb{C}^{N\times N}$ with $\mathcal{B}|0\rangle = |b\rangle$ and potentially asymptotic time complexity $T_{\mathcal{B}}$, we can approximate the phase of the associated eigenvalue of $|b\rangle$ using t bits with $\Theta(T_{\mathcal{B}} + t^2)$ operations and a success probability of at least $1-\varepsilon$.

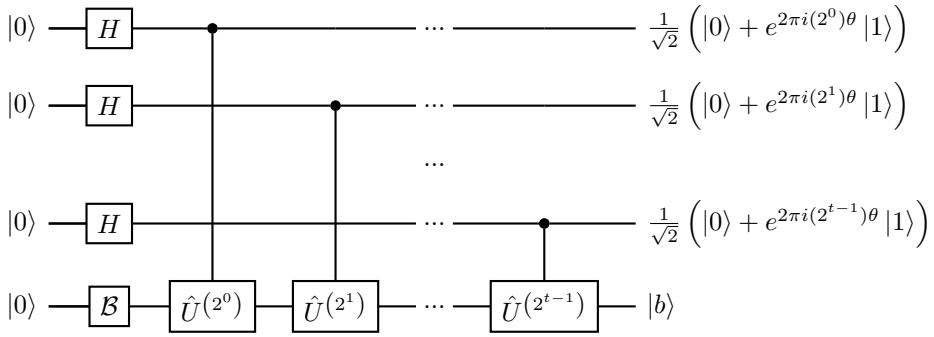


Figure 5: Circuit diagram for the first part of the general QPE algorithm. The $t \in \mathbb{N}_{\geq 1}$ qubits are used to approximate a binary representation of the eigenvalue phase, as we can see on the right. The essential point of the first part is to store the vector $\bigotimes_{k=0}^{2^t-1}\left(|0\rangle + e^{2\pi i(2^k)\theta}|1\rangle\right)|b\rangle = \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle U^j|b\rangle$, as one can recognize by aligning the binary representation of the summed up factor in the amplitude exponent with the canonical state for each possible product taken. Replication of [7, pp. 221-226].

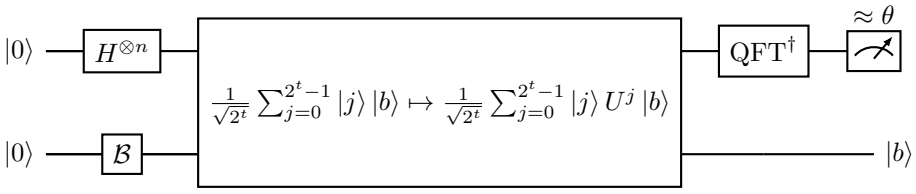


Figure 6: Circuit diagram for the general QPE algorithm.

Remark 2.29. Consider, that

- this subsection was designated to showcase the most commonly seen QPE algorithm, but it will be of no further interest for the remaining thesis, as we will use different custom routines for phase estimation later on.
- the algorithm stores the approximation in a register to be measured, which may make it unsuitable for a subroutine in a larger algorithm. Furthermore, the requirement for the existence of the controlled exponential unitaries of U may be very restrictive.

2.7 Hamiltonian Simulation

Consider our initial discussion on QM in Section 1.1. The time postulate in our form of interest was given in Equation (1.1.2). Since for a unitary $H \in \mathbb{C}^{n\times n}$ and a time $t \in \mathbb{R}_{\geq 0}$, we have, that e^{iHt} is unitary via Theorem 1.18, we can speak of particles, which evolve according to such a matrix exponential of a Hamiltonian, so for a quantum particle with state $|\psi(\cdot)\rangle$, we could have

$$|\psi(t)\rangle = e^{iHt} |\psi(0)\rangle \tag{2.7.1}$$

for one such time t. In the context of qubits, we may reformulate this fact as the following problem.

Problem 2.30. Let $H \in \mathbb{C}^{N\times N}$, $N \coloneqq 2^n$, be Hermitian and $t \in \mathbb{R}_{\geq 0}$. The problem of applying the unitary operator e^{iHt} to a state $|\psi\rangle \in \mathbb{C}^N$ is called *Hamiltonian simulation*.

This problem is in a very active state of research and has yielded a lot of results over the years, some notable mentions are [30, 31]. Especially the Hamiltonian simulation with Hermitian operators in infinite-dimensional Hilbert spaces is of interest for general particle physics, QM and also for Quantum Chemistry, as it allows the simulation of the development of molecules, which in turn has multiple applications like the development of pharmaceuticals [32, pp. 14-18][6]. We are especially interested in a comparatively old result by Berry et al. [33], which draws results from the same paper we referenced for the problem of quantum state generation by Aharonov et al. [22].

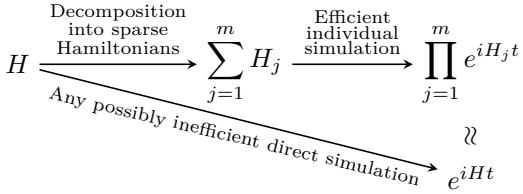
One approach to Problem 2.30 may be to approximate e^{iHt} directly by the associated Taylor series as in Definition 1.16, but such an approach has been proven to not be satisfactory in practice [7, p. 206], although some voices [31, p. 1] argue, that this commonly presented perspective is too pessimistic. Here, we consider techniques based on decomposing the Hamiltonian into a sum of Hamiltonians $H = \sum_{j=1}^{m} H_j$ and then individually simulating $H_1, ..., H_m$ for $m \in \mathbb{N}_{\geq 1}$. The main problem of using this idea directly is that the Hamiltonians may not commute, which violates Lemma 1.17. Instead, we compute an approximation of the so-called *Trotter formula*.

Theorem 2.31 (Trotter Formula)**.** For Hamiltonians $A, B \in \mathbb{C}^{n\times n}$ and $t \in \mathbb{R}_{\geq 0}$, we have

$$e^{i(A+B)t} = \lim_{m\to\infty} (e^{iAt/m} e^{iBt/m})^m \tag{2.7.2}$$

Note that we explicitely do not require commuting operators here. We can also omit i and t, but it suits our context. The proof is can be found in [7, p. 207]. Writing a sum of Hamiltonians in such a form is also the basic idea of the constructions used by Berry et al..

We now consider sparse Hamiltonians, according to Definition 1.27. Let $H \in \mathbb{C}^n$ be an s-sparse, efficiently row computable Hamiltonian, where $s \in \mathbb{N}$. We first decompose the Hamiltonian into easily simulatable Hamiltonians, and then apply a recursion formula found by the researcher Suzuki, which is in turn based on the Trotter formula. As each Hamiltonian is easily simulatable, and we have approximately equality, we obtain the required simulation. The idea can be illustrated via the following sketch, which resembles a commutative diagram.



Definition 2.32. The *iterated logarithm* is defined as

$$\log_2^*\colon \mathbb{R}_{>0} \to \mathbb{N}, x \mapsto \begin{cases} 0 & x \leq 1 \\ \min\left\{ i \in \mathbb{N}_{\geq 1} \;\middle|\; (\underbrace{\log_2 \circ ... \circ \log_2}_{i \text{ times}})(x) \leq 1 \right\} & x > 1 \end{cases} \tag{2.7.3}$$

[6]Note, that the Hamiltonians are approximated in a real setting.

Example 2.33. Whilst the iterated logarithm is a monotonically increasing function with discrete values, it grows incredibly slow. A textbook example is $\log_2^*(2^{65536}) = 5$, which is a problem size, that is much greater than 10^{80}, the approximate number of atoms in the observable universe [34, pp. 58-59].

We first have the following theorem.

Theorem 2.34. There is a decomposition of H of form $H = \sum_{i=1}^{6s^2} H_i$, s.t. for each $i \in [1, 6s^2]_{\mathbb{N}}$, $H_i \in \mathbb{C}^{n\times n}$ is a 1-sparse Hamiltonian, requiring an access time of $\mathcal{O}(\log^*(n))$ to H to determine its at most one coefficient in one row.

Proof idea. The proof is based on a combinatorial argument on the entries of H. We consider the graph $G_H := ([1,n]_{\mathbb{N}}, E_H)$, where $(i,j) \in E_H$ iff $H_{ij} \neq 0$. Since $H_{ij} = H_{ji}^*$, which one checks via Definition 1.7, we can take the graph to be undirected. We illustrate this in Figure 7 for an example. A graph coloring is obtained and to prevent duplicate edges due to the coloring predicate, an additional parameter, determined by so-called *deterministic coin-tossing*, is introduced. We will not describe the method here. For each pair (i,j) and each additional such parameter ν, we introduce a Hamiltonian, totalling $6s^2$ Hamiltonians, as we only consider such (i,j), where $H_{ij} \neq 0$, as well as only six possible values for ν according to the labeling scheme. The argument for the choice of ν is one of the main contributions of that proof and contains an arguably long case distinction. With a small illustration, it can be found in [33, p. 6-8].

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

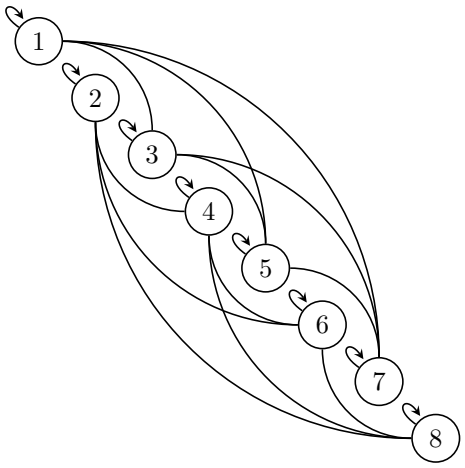


Figure 7: The described graph for the chess-pattern Hamiltonian $(\sum_{j=0}^{1}\sum_{k=0}^{1} |j\rangle\langle k|)^{\otimes 2} \otimes E_2$.

We further take the following theorem as given. It concerns the Hamiltonian simulation of a decomposable Hamiltonian.

Theorem 2.35. Let a decomposed Hamiltonian $H = \sum_{j=1}^{m} H_j \in \mathbb{C}^{n\times n}$ with Hamiltonians $H_1, ..., H_m \in \mathbb{C}^{n\times n}$ be given. Define the *Suzuki higher order integrators* S_{2k} *of order* $k \in \mathbb{N}_{\geq 1}$ recursively via

$$S_2(\lambda) := \prod_{j=1}^{m} e^{H_j\lambda/2} \prod_{j=1}^{m} e^{H_{m-j+1}\lambda/2} \qquad S_{2k}(\lambda) := S_{2(k-1)}(p_k\lambda)^2 S_{2(k-1)}((1-4p_k)\lambda) S_{2(k-1)}(p_k\lambda)^2 \tag{2.7.4}$$

where $\lambda \in \mathbb{C}$, $p_k := 1/(4 - 4^{1/(2k-1)})$. We have

$$\|\exp(\lambda H) - S_{2k}(\lambda/r)^r\| \in \mathcal{O}(|\lambda|^{2k+1}/r^{2k}) \tag{2.7.5}$$

for $r \in \mathbb{N}_{\geq 1}$.

Berry et al. [33] cite this result from [35]. Precisely, their cited result in eq. (4) of the paper is an application of the form in the eqs. (40-42) from [35, p. 4]. The paper itself builds up on several research

results on quantum monte carlo simulations [35, p. 1] and concerns general decompositions of exponential operators. In the case of Hamiltonian simulation, we let $\lambda \coloneqq it$. Using this bound and the ideas for the decomposition mentioned, the authors then obtain the following result.

Theorem 2.36. Using Theorem 2.35 and Theorem 2.34, there is a quantum algorithm, that computes the Hamiltonian simulation of a s-sparse, $s \in \mathbb{N}$, efficiently-row computable Hamiltonian H for a time $t \in \mathbb{R}_{\geq 0}$ acting on n qubits in time

$$\mathcal{O}\left(n \log_2^*(n)^2 s^4 \|H\| t e^{2\sqrt{\ln(5)\ln(s^2\|H\|t/\varepsilon)}}\right) = \tilde{\mathcal{O}}(\log_2(N) s^4 t) \tag{2.7.6}$$

where we denote by $\tilde{\mathcal{O}}(\cdot)$ the runtime under negligence of the expression $\log_2^*(n)^2 \|H\| e^{2\sqrt{\ln(5)\ln(s^2\|H\|t/\varepsilon)}}$.

With the simplified expression for the runtime using the notation $\tilde{\mathcal{O}}(\cdot)$, we follow Harrow et al. [3, pp. 5-6].

Derivation. We give a short derivation of this result from the results of the paper. In [33, pp. 8-9], we have the algorithm runtime with auxiliary operations of

$$\mathcal{O}(n \log_2^*(n)^2 d^2 5^{2k} (d^2\tau)^{1+1/(2k)}/\varepsilon^{1/(2k)}) = \mathcal{O}(n \log_2^*(n)^2 d^4 \tau 5^{2k} (d^2\tau)^{1/(2k)}/\varepsilon^{1/(2k)}) \tag{2.7.7}$$

with $\tau \coloneqq \|H\|t$ and $d \coloneqq s$ from the notation of the paper. The parameter $k \in \mathbb{N}$ can be chosen at will, it corresponds to the depth of the recursion of the Suzuki higher order integrators. To optimize k, we consider the given minimum at [33, pp. 1-2]. To derive it, observe

$$5^{2k}(d^2\tau/\varepsilon)^{1/(2k)} = e^{2k\ln(5) + \ln\left(d^2\tau/\varepsilon\right)/(2k)} \tag{2.7.8}$$

and let

$$f\colon \mathbb{R} \to \mathbb{R}, k \mapsto 2k\ln(5) + \ln\left(d^2\tau/\varepsilon\right)/(2k) \tag{2.7.9}$$

reusing the symbol k. Then we have

$$f'(k) = 2\ln(5) - \frac{\ln\left(d^2\tau/\varepsilon\right)}{2k^2}, f''(k) = \frac{\ln\left(d^2\tau/\varepsilon\right)}{k^3} \tag{2.7.10}$$

Solving for a minimum and using $\log_5(x) = \ln(x)/\ln(5)$ for any $x \in \mathbb{R}_{>0}$ thus gives

$$k \approx \sqrt{\frac{\ln(d^2\tau/\varepsilon)}{4\ln(5)}} = \frac{1}{2}\sqrt{\log_5(d^2\tau/\varepsilon)} \tag{2.7.11}$$

Plugging this into Equation (2.7.8) then gives the value

$$e^{\ln(5)\sqrt{\log_5(d^2\tau/\varepsilon)} + \sqrt{\ln(5)}\sqrt{\ln(d^2\tau/\varepsilon)}} = e^{2\sqrt{\ln(5)\ln(d^2\tau/\varepsilon)}} \tag{2.7.12}$$

So the runtime is

$$\mathcal{O}\left(n \log_2^*(n)^2 s^4 \|H\| t e^{2\sqrt{\ln(5)\ln(s^2\|H\|t/\varepsilon)}}\right) \tag{2.7.13}$$

Remark 2.37. The contribution of keeping the error gap to the runtime is sublinear, meaning, that $e^{2\sqrt{\ln(5)\ln(s^2\|H\|t/\varepsilon)}} \in o(1/\varepsilon)$. Consider for that, that in general $e^{2\sqrt{\ln(1/\varepsilon)}} \in o(1/\varepsilon)$, as $\lim_{\varepsilon\to\infty} e^{2\sqrt{\varepsilon}}/\varepsilon = \lim_{\varepsilon\to\infty} e^{2\sqrt{\varepsilon}(1-(1/2)\sqrt{\varepsilon})} \to 0$. The error contribution to the runtime thus may be neglected.

With this result, we may also introduce another auxiliary gate, taken from [3, pp. 3-4].

Definition 2.38. The *conditional Hamiltonian evolution* gate for some Hermitian $A \in \mathbb{C}^{N\times N}, T \coloneqq 2^t, t \in \mathbb{N}_{\geq 1}, N \coloneqq 2^n$ and $t_0 \in \mathbb{R}_{>0}$ is the unitary map:

$$\mathrm{CHE}_{T,N,A,t_0}\colon \mathbb{C}^{TN} \to \mathbb{C}^{TN}, |x\rangle|y\rangle \mapsto \left(\sum_{\tau=0}^{T-1} |\tau\rangle\langle\tau| \otimes e^{iA\tau t_0/T}\right)|x\rangle|y\rangle \tag{2.7.14}$$

$$\mathrm{CHE}_{T,N,A,t_0} = \begin{pmatrix} e^{iA\cdot 0\cdot t_0/T} & 0 & \dots & 0 \\ 0 & e^{iA\cdot 1\cdot t_0/T} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & e^{iA\cdot (T-1)\cdot t_0/T} \end{pmatrix} \tag{2.7.15}$$

With this way of writing out the matrix for the conditional Hamiltonian evolution, it also becomes clear that it is unitary, as we can use Lemma 1.17 and simulate the evolution for each $k \in [0, N-1]_{\mathbb{N}}$th of the n qubits individually using A with time $t_0 k/T$, and thus a valid quantum gate.

Remark 2.39. We will regard the runtime of the controlled Hamiltonian evolution to be the same as the individual Hamiltonian simulation, as in Theorem 2.36.

3 The HHL Algorithm

The currently asymptotically best classical method known for solving SLEs is the conjugate gradient method, which runs in worst case time $\mathcal{O}(Ns\sqrt{\kappa}\log_2(1/\varepsilon))$, as described in [36, complexity analysis on pp. 37-38] or [11, pp. 279-306]. Here, $\varepsilon \in \mathbb{R}_{>0}$ is the error cap, $\kappa \in \mathbb{R}_{\geq 1}$ the condition number of the input $N \times N$ matrix, where $N \in \mathbb{N}_{\geq 1}$, and $s \in \mathbb{N}$ is the sparsity as in Definition 1.27, assuming that the matrix is efficiently row-computable. In many practical cases, the given matrix has about $\mathcal{O}(\sqrt{N})$ non-zero entries, for which the algorithm yields a runtime of about $\mathcal{O}(N^{3/2}\sqrt{\kappa})$, ignoring the complexity factor for keeping the error low. In 2008, the researchers Harrow, Hassidim and Lloyd (HHL) described a quantum algorithmic approach to solving SLEs. In this section, we will give a rigorous description of the HHL algorithm, along with a rework of the original analysis. The original formulation can be found in [3]. We will draw a lot of information from the alternative, more comprehensive formulation presented by Dervovic et al. in [12, pp. 28-42] as well. Our contribution lies in the explicit description of the auxiliary procedures, such as the initialization of a helper state and a description on three-dimensional rotations, as well as more explicit runtime and error bounds. Most of these helper algorithms have been elaborated in Section 2.

3.1 Problem Description and Assumptions

Starting off, let us pay attention to the general problem of solving SLEs.

Problem 3.1. Given $A \in \mathbb{C}^{m\times n}, b \in \mathbb{C}^m$ with $m, n \in \mathbb{N}_{\geq 1}$, find an $x \in \mathbb{C}^n$ with $Ax = b$, if it exists.

The HHL algorithm, in its original formulation, has the following requirements:

1. $m = n$ and n is a power of two.
2. $\|b\| = 1$, so we may write $|b\rangle$.
3. $|b\rangle$ can be initialized efficiently in a $\log_2(n)$-qubit register.
4. A is well-conditioned, as in Definition 1.28.
5. The condition number $\kappa(A) \in \mathbb{R}_{\geq 1}$, using the notation from Definition 1.23, or at least an upper bound of it, must be known in advance.

The result is then stored in a $\log_2(n)$-qubit register. So we may reformulate the problem.

Problem 3.2. Given a well-conditioned $A \in \mathbb{C}^{N\times N}$ with condition number $\kappa := \kappa(A) \in \mathbb{R}_{\geq 1}$ and an efficiently initializable state $|b\rangle \in \mathbb{C}^N$, $N := 2^n$ and $n \in \mathbb{N}_{\geq 1}$, find or approximate a quantum state $|x\rangle \in \mathbb{C}^N$, s.t. there is a $C \in (0, \infty)$ with $A(C|x\rangle) = |b\rangle$.

Note, that A may not be isometric, so we require that $|x\rangle$ solves the inversion problem up to some positive multiplicative constant, which we can recover by comparing two non-zero elements of the vectors $A|x\rangle$ and $|b\rangle$. In Section 3.4, we will discuss some techniques for relaxing the assumptions.

3.2 Overview

We carry over the notation introduced in Problem 3.2. Let furthermore $s \in \mathbb{N}$ be the sparsity of A and let $\kappa := \kappa(A)$. In total, we will require $t + n + 1$ qubits, where $t \in \mathbb{N}_{\geq 5}$ is a hyperparameter. Let $T := 2^t$. One will need to choose t appropriately for the matrix A, as described in the second next paragraph. In Dervovic et al., the first register is called the *clock* register, the second the *input* register and the third is an auxiliary qutrit [12, p. 30]. For the following, let $\{(\lambda_1, |v_1\rangle), ..., (\lambda_N, |v_N\rangle)\} \subseteq \left[\frac{1}{\kappa}, 1\right] \times \mathbb{C}^N$ be the eigenvalue-eigenvector pairs of an eigenbasis of A and decompose $|b\rangle$ as

$$|b\rangle =: \sum_{j=1}^{N} \beta_j |v_j\rangle = \sum_{j=1}^{N} \langle b|v_j\rangle |v_j\rangle \tag{3.2.1}$$

Brief Sketch of the Algorithm

We now give a brief sketch of the algorithm. Following the decomposition of $|b\rangle$ and Corollary 1.12, the aim is to approximate the quantum state

$$|x\rangle = \frac{1}{C}\sum_{j=1}^{N}\frac{\beta_j}{\lambda_j}|v_j\rangle \tag{3.2.2}$$

with some $C \in \mathbb{R}_{>0}$. The algorithm starts in the state $|0\rangle|0\rangle|0\rangle \in \mathbb{C}^{TN\cdot 3}$. The first steps are aimed towards approximating all eigenvalues of A simultaneously from all T possible canonical state vectors for the first register. To be precise, a state of form

$$\sum_{j=1}^{N}\beta_j|\tilde{\lambda_j}\rangle|v_j\rangle|0\rangle \tag{3.2.3}$$

is first produced. $\tilde{\lambda_j}$ here represents an approximation of λ_j, s.t. $|\lambda_j - \tilde{\lambda_j}| < \frac{2\pi}{t_0}$, where $t_0 \in \mathbb{R}_{>0}$ is later chosen to minimize the overall error, but it is a large value in general. Such an approximation exists, if t and t_0 are chosen well. The qutrit is then rotated to yield the state

$$\sum_{j=1}^{N}\beta_j|\tilde{\lambda_j}\rangle|v_j\rangle\left(\sqrt{1-f^2(\tilde{\lambda_j})-g^2(\tilde{\lambda_j})}|0\rangle + f(\tilde{\lambda_j})|1\rangle + g(\tilde{\lambda_j})|2\rangle\right) \tag{3.2.4}$$

The functions $f\colon \mathbb{R}_{>0} \to [0,1]$, $g\colon \mathbb{R}_{>0} \to [0,1]$ are so-called *filter functions* and defined in a way that allows filtering out tiny eigenvalues, such that taking their reciprocal does not produce an inaccurate state. With the assumptions made in Section 3.1, $g(\tilde{\lambda_j}) \approx 0$ and the filter functions thus evaluate to approximately give the state

$$\sum_{j=1}^{N}\beta_j|\tilde{\lambda_j}\rangle|v_j\rangle\left(\sqrt{1-\frac{1}{4\kappa^2\tilde{\lambda_j}^2}}|0\rangle + \frac{1}{2\kappa\tilde{\lambda_j}}|1\rangle\right) \tag{3.2.5}$$

Uncomputing the eigenvalue approximation in the first register and applying amplitude amplification, as in Algorithm 1, to measure a 1 in the qutrit gives us the state

$$\frac{1}{\sqrt{\frac{1}{4\kappa^2}\sum_{j=1}^{N}\frac{|\beta_j|^2}{\tilde{\lambda_j}^2}}}\frac{1}{2\kappa}\sum_{j=1}^{N}\frac{\beta_j}{\tilde{\lambda_j}}|v_j\rangle \tag{3.2.6}$$

in the second register, which corresponds to our target state as in Equation (3.2.2).

Description of the Entire Algorithm

We now give a full description of the HHL algorithm with an associated circuit diagram. The choice of the parameters ε, t_0 and t will be explained in the analysis of the algorithm.

Algorithm 2 HHL Algorithm

Given: A well-conditioned $A \in \mathbb{C}^{N\times N}$ with condition number $\kappa \in \mathbb{R}_{\geq 1}$ or an upper bound of it, where $N \coloneqq 2^n$ with $n \in \mathbb{N}_{\geq 1}$, a vector $|b\rangle \in \mathbb{C}^N$, an efficiently implementable unitary $\mathcal{B} \in \mathbb{C}^{N\times N}$ with $\mathcal{B}|0\rangle = |b\rangle$ and an error cap $\varepsilon \in \left(0, \frac{100}{4\pi}\right)$.

Return: A quantum state $|\tilde{x}\rangle \in \mathbb{C}^N$ with $\||x\rangle - |\tilde{x}\rangle\| \leq \varepsilon$, where $|x\rangle$ corresponds to the normalization of a vector $x \in \mathbb{C}^N$ with $Ax = |b\rangle$.

1: Let $t_0 \coloneqq 200\frac{\kappa}{\varepsilon}$ and $t \coloneqq \max\{\lceil \log_2(t_0/2\pi) + 1\rceil, 5\}$.
2: $|\Psi\rangle \coloneqq |0\rangle|0\rangle|0\rangle \in \mathbb{C}^{TN\cdot 3}$ empty $t+n$ qubit-register with an ancilla qutrit.
3: $|\Psi\rangle \leftarrow (\mathcal{T}\otimes\mathcal{B}\otimes E_3)|\Psi\rangle$
4: $|\Psi\rangle \leftarrow (\mathrm{CHE}_{T,N,A,t_0}\otimes E_3)|\Psi\rangle$
5: $|\Psi\rangle \leftarrow (\mathrm{QFT}_T^\dagger \otimes E_{T\cdot 3})|\Psi\rangle$
6: $|\Psi\rangle \leftarrow \mathcal{R}|\Psi\rangle$ with $\mathcal{R}$ as defined below.
7: $|\Psi\rangle \leftarrow (\mathrm{QFT}_T \otimes E_{T\cdot 3})|\Psi\rangle$
8: $|\Psi\rangle \leftarrow (\mathrm{CHE}_{T,N,A,t_0}^\dagger \otimes E_3)|\Psi\rangle$
9: $|\Psi\rangle \leftarrow (\mathcal{T}^\dagger \otimes E_{N\cdot 3})|\Psi\rangle$
10: Perform amplitude amplification using Algorithm 1 on the previous steps to measure a 1 for the state $|\Psi\rangle$ obtained from the previous steps with the function $\chi\colon \{0,1\}^{TN\cdot 3} \to \{0,1\}$, s.t. $\chi(x,y,z) = 1$, iff $z = 1$ for any $x \in [0, T-1]_{\mathbb{N}}$ and $y \in [0, N-1]_{\mathbb{N}}$.
11: **return** the n-qubit register of $|\Psi\rangle$.

The unitaries $\mathcal{T} \in \mathbb{C}^{T\times T}$ and $\mathcal{B} \in \mathbb{C}^{N\times N}$ are characterized by the following actions on $|0\rangle$:

$$\mathcal{T}|0\rangle = \sqrt{\frac{2}{T}}\sum_{\tau=0}^{T-1}\sin\left(\frac{\pi\left(\tau+\frac{1}{2}\right)}{T}\right)|\tau\rangle \qquad \mathcal{B}|0\rangle = |b\rangle \tag{3.2.7}$$

We give a more detailled description and the implementation of $\mathcal{T}$ further below. Furthermore, the so-called *filter functions* $f, g\colon \mathbb{R}_{\geq 0} \to \left[0, \frac{1}{2}\right]$ and their associated qutrit rotation unitary shall be defined as

$$f_\kappa(\lambda) \coloneqq \begin{cases} 0 & \lambda < \frac{1}{2\kappa} \\ \frac{1}{2}\sin\left(\frac{\pi}{2}\cdot\frac{\lambda-\frac{1}{2\kappa}}{\frac{1}{\kappa}-\frac{1}{2\kappa}}\right) & \frac{1}{2\kappa} \leq \lambda < \frac{1}{\kappa} \\ \frac{1}{2\kappa\lambda} & \frac{1}{\kappa} \leq \lambda \end{cases} \qquad g_\kappa(\lambda) \coloneqq \begin{cases} \frac{1}{2} & \lambda < \frac{1}{2\kappa} \\ \frac{1}{2}\cos\left(\frac{\pi}{2}\cdot\frac{\lambda-\frac{1}{2\kappa}}{\frac{1}{\kappa}-\frac{1}{2\kappa}}\right) & \frac{1}{2\kappa} \leq \lambda < \frac{1}{\kappa} \\ 0 & \frac{1}{\kappa} \leq \lambda \end{cases} \tag{3.2.8}$$

$$\mathcal{R} \coloneqq \sum_{\theta=0}^{T-1}|\theta\rangle\langle\theta| \otimes E_N \otimes R\left(\arcsin\left(\frac{f\left(\frac{2\pi\theta}{t_0}\right)}{\sqrt{1-g^2\left(\frac{2\pi\theta}{t_0}\right)}}\right), \arcsin\left(g\left(\frac{2\pi\theta}{t_0}\right)\right)\right) \tag{3.2.9}$$

where $\mathcal{R}$ is constructed as in the proof of Theorem 2.17, here without additional helper qubits. Let furthermore $f \coloneqq f_\kappa$ and $g \coloneqq g_\kappa$, as κ is fixed.

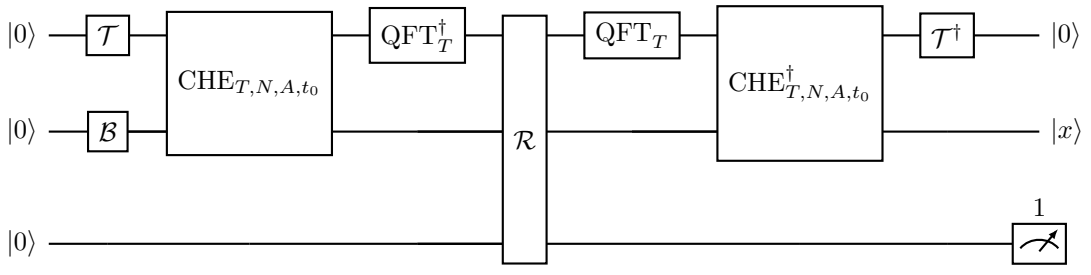


Figure 8: Circuit diagram for the HHL algorithm. On the right, the register states for a perfect result are presented. We measure a 1, indicating a good result. We have not illustrated the amplitude amplification.

Initialization Procedures

As explained, we require two procedures $\mathcal{T}$ and $\mathcal{B}$ to prepare the first two registers of t and n qubits respectively, which we will explain in this paragraph starting with $\mathcal{T}$. At first sight, it is not obvious, that the condition in Equation (3.2.7) results in a valid quantum state. We introduce the following lemma.

Lemma 3.3. For any $x \in \mathbb{C}$ it holds that:

$$\sin^2\left(\frac{x}{2}\right) = \frac{1-\cos(x)}{2} \tag{3.2.10}$$

Proof. We use Theorem B.4 and Theorem B.3 to obtain:

$$\cos(2x) = \cos^2(x) - \sin^2(x) = 1 - 2\sin^2(x) \tag{3.2.11}$$

Solving after $\sin^2(x)$ and substituting x for $x/2$ yields the statement. ■

The next lemma then confirms the claim from before.

Lemma 3.4. Let $T \coloneqq 2^t$, $t \in \mathbb{N}_{\geq 1}$ and $\tau \in [0, T-1]_{\mathbb{N}}$. It holds, that

$$\sum_{k=0}^{\tau} \frac{2}{T} \sin^2\left(\frac{\pi(k+\frac{1}{2})}{T}\right) = \frac{1}{T}\left(\tau + 1 - \frac{\sin\left(2(\tau+1)\frac{\pi}{T}\right)}{2\sin\left(\frac{\pi}{T}\right)}\right) \tag{3.2.12}$$

Proof. We can use Lemma 2.22 from Section 2.5. Consider for any $k \in [0, \tau]_{\mathbb{N}}$, using Lemma 3.3:

$$\sin^2\left(\frac{\pi(k+\frac{1}{2})}{T}\right) = \sin^2\left((2k+1)\frac{\pi}{2T}\right) = \frac{1}{2}\left(1 - \cos\left((2k+1)\frac{\pi}{T}\right)\right) \tag{3.2.13}$$

So

$$\sum_{k=0}^{\tau} \frac{2}{T} \sin^2\left(\frac{\pi(k+\frac{1}{2})}{T}\right) = \frac{1}{T}\left(\tau + 1 - \sum_{k=0}^{\tau} \cos\left((2k+1)\frac{\pi}{T}\right)\right) \tag{3.2.14}$$

$$= \frac{1}{T}\left(\tau + 1 - \frac{\sin\left(2(\tau+1)\frac{\pi}{T}\right)}{2\sin\left(\frac{\pi}{T}\right)}\right) \tag{3.2.15}$$

under the use of Lemma 2.22. ■

Inserting $\tau = T - 1$ gives the claim, that $\mathcal{T}$ gives a valid quantum state. This closed formula further gives the following claim.

Theorem 3.5. The procedure $\mathcal{T}$ can be implemented to run in time $\mathcal{O}(t)$ with arbitrary precision.

Proof. We wish to give an antiderivative P of a probability density function $p\colon [0,\pi] \mapsto [0,1]$ with for any $\tau \in [0, T-1]_{\mathbb{N}}$

$$\int_{x_L^{t,\tau}}^{x_R^{t,\tau}} p = \frac{2}{T}\sin^2\left(\frac{\pi(\tau+\frac{1}{2})}{T}\right) \tag{3.2.16}$$

where $x_L^{t,\tau} \coloneqq \pi\frac{\tau}{T}, x_R^{t,\tau} \coloneqq \pi\frac{\tau+1}{T}$. Let

$$P\colon [0,\pi] \to [0,1], x \mapsto \int_0^x p \tag{3.2.17}$$

Then for any $\tau \in [0, T-1]_{\mathbb{N}}$:

$$P(x_R^{t,\tau}) = \int_0^{x_R^{t,\tau}} p = \sum_{k=0}^{\tau} \frac{2}{T}\sin^2\left(\frac{\pi(k+\frac{1}{2})}{T}\right) = \frac{1}{T}\left(\tau + 1 - \frac{\sin\left(2(\tau+1)\frac{\pi}{T}\right)}{2\sin\left(\frac{\pi}{T}\right)}\right) \tag{3.2.18}$$

using Lemma 3.4. Consider the equation $x = \frac{\pi(\tau+1)}{T}$, from which we get $\tau = \frac{T}{\pi}x - 1$ to substitute τ in Equation (3.2.18). Letting x be loose gives

$$P(x) = \frac{1}{T}\left(\frac{T}{\pi}x - \frac{\sin(2x)}{2\sin\left(\frac{\pi}{T}\right)}\right) \tag{3.2.19}$$

from which we obtain, that p is efficiently integrable, as efficient approximations of the sine exist. The use of Theorem 2.7 gives the claim. ■

Remark 3.6. The values of Equation (3.2.16) resemble the normal distribution, as Figure 9 shows. Furthermore, p roughly resembles a sigmoidal function, as the cumulative sums of the values in Equation (3.2.16) show, as pictured in Figure 10. Especially the first observation may give some insight into why these coefficients were chosen, although we have not yet made the connection to the error analysis.

Remark 3.7. The problem of recovering an efficiently integrable probability density function, or its associated integral function, from given definite integrals may be interesting for the general quantum algorithmic toolbox. The proof of Theorem 3.5 performs such a task for a very specific example, where much trigonometric structure is indeed involved, but finding such simple expressions may be hard in general. Uncountably many functions may suffice for such a task and canonical continuous functions for such interpolation tasks exist, one may look at Lagrangian interpolation, see [17, p. 192], but it is still unclear, if canonical efficiently computable functions for such interpolations exist.

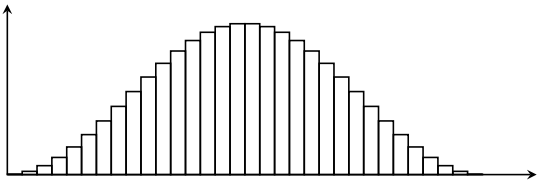


Figure 9: Sketch of the amplitudes, here for $t = 5$ and scaled by 16.

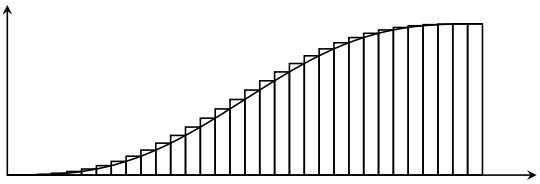


Figure 10: Sketch of the cumulative amplitude sums, here for $t = 5$ and scaled by 1. The associated integral function of the probability distribution p, P, as found in the proof of Theorem 3.5, is also depicted.

As for the procedure $\mathcal{B}$: We have discussed the problem of quantum state generation in Section 2.2. Efficient state generation is a key issue here, as an inefficient state generation procedure will drown the runtime of the HHL algorithm, as can be seen in Algorithm 2.

3.3 Analysis for Well-Conditioned Matrices

We present the analysis from [3] with slightly different constants in the results. We assume, that none of the subprocedures produce an error, which one may consider to be a reasonable assumption due to our previous discussion on the complexity for keeping a low error gap for each subprocedure. Consider again the assumptions made in Section 3.1.

First Steps

We first follow along the description of the algorithm in Algorithm 2 and observe the change of the registers. Initializing the first two registers yields

$$|0\rangle|0\rangle|0\rangle \xmapsto{\mathcal{T}\otimes\mathcal{B}\otimes E_3} \sqrt{\frac{2}{T}}\sum_{\tau=0}^{T-1}\sin\left(\frac{\pi(\tau+\frac{1}{2})}{T}\right)|\tau\rangle|b\rangle|0\rangle = \sqrt{\frac{2}{T}}\sum_{j=1}^{N}\beta_j\sum_{\tau=0}^{T-1}\sin\left(\frac{\pi(\tau+\frac{1}{2})}{T}\right)|\tau\rangle|v_j\rangle|0\rangle \tag{3.3.1}$$

After that, applying $\mathrm{CHE}_{T,N,A,t_0} \otimes E_3$ with effect on the first two registers and the use of Theorem 1.21 gives

$$\sqrt{\frac{2}{T}} \sum_{j=1}^{N} \beta_j \sum_{\tau=0}^{T-1} \sin\left(\frac{\pi(\tau+\frac{1}{2})}{T}\right) \mathrm{CHE}_{T,N,A,t_0} |\tau\rangle |v_j\rangle |0\rangle \tag{3.3.2}$$

$$= \sqrt{\frac{2}{T}} \sum_{j=1}^{N} \beta_j \sum_{\tau=0}^{T-1} \sin\left(\frac{\pi(\tau+\frac{1}{2})}{T}\right) e^{i\lambda_j \tau t_0/T} |\tau\rangle |v_j\rangle |0\rangle \tag{3.3.3}$$

Now we apply $\mathrm{QFT}_T^\dagger \otimes E_{N\cdot 3}$, which, after some reordering, results in

$$\sqrt{\frac{2}{T}} \sum_{j=1}^{N} \beta_j \sum_{\tau=0}^{T-1} \sin\left(\frac{\pi(\tau+\frac{1}{2})}{T}\right) e^{i\lambda_j \tau t_0/T} \mathrm{QFT}_T^\dagger |\tau\rangle |v_j\rangle |0\rangle \tag{3.3.4}$$

$$= \frac{\sqrt{2}}{T} \sum_{j=1}^{N} \beta_j \sum_{\tau=0}^{T-1} \sin\left(\frac{\pi(\tau+\frac{1}{2})}{T}\right) e^{i\lambda_j \tau t_0/T} \sum_{k=0}^{T-1} e^{-2\pi i k\tau/T} |k\rangle |v_j\rangle |0\rangle \tag{3.3.5}$$

$$= \frac{\sqrt{2}}{T} \sum_{j=1}^{N} \beta_j \sum_{k=0}^{T-1} \left(\sum_{\tau=0}^{T-1} \sin\left(\frac{\pi(\tau+\frac{1}{2})}{T}\right) e^{\frac{i\tau}{T}(\lambda_j t_0 - 2\pi k)} \right) |k\rangle |v_j\rangle |0\rangle \tag{3.3.6}$$

$$\overset{(1)}{=} \sum_{j=1}^{N} \beta_j \sum_{k=0}^{T-1} \alpha_{j,k} |k\rangle |v_j\rangle |0\rangle \tag{3.3.7}$$

(1) Note that we define for these indices j, k the values $\alpha_{j,k} \in \mathbb{C}$ and $\delta_{j,k} \in \mathbb{R}$ via

$$\alpha_{j,k} \coloneqq \frac{\sqrt{2}}{T} \sum_{\tau=0}^{T-1} \sin\left(\frac{\pi(\tau+\frac{1}{2})}{T}\right) e^{\frac{i\tau}{T}\delta_{j,k}} \qquad \delta_{j,k} \coloneqq \lambda_j t_0 - 2\pi k \tag{3.3.8}$$

This intermediate result corresponds to a "good" approximation of the eigenvalues of A, as we will prove in the following paragraph. We only make one small observation.

Observation 3.8. We have $\sum_{k=0}^{T-1} |\alpha_{j,k}|^2 = 1$. This follows by considering $|b\rangle = |v_j\rangle$ and thus having

$$\sum_{j=1}^{N} \beta_j \sum_{k=0}^{T-1} \alpha_{j,k} |k\rangle |v_j\rangle |0\rangle = \sum_{k=0}^{T-1} \alpha_{j,k} |k\rangle |v_j\rangle |0\rangle \tag{3.3.9}$$

be a valid quantum state with the $\alpha_{j,k}$ values being independent of β_j. Furthermore, we have $|\alpha_{j,k}| \leq 1$.

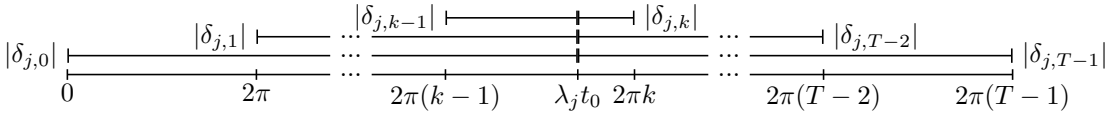


Figure 11: A line representing $[0, 2\pi(T-1)]$ with marks for understanding the behavior of the approximations for one $\delta_{j,k}$ value. We assume an appropriate choice for t, as described in this text. In this case, the approximation seems to be of poor quality, increasing t will improve the accuracy as then the interval $[2\pi(k-1), 2\pi k]$ will be split in half and $2\pi(2k-1)$ will give a better approximation.

Analysis of the Phase Estimation

The following analysis of the coefficients $\alpha_{j,k}$ is based on the original HHL paper [3, pp. 10-11] and Dervovic et al. [12, pp. 32-33], but we do not fully agree with the assumptions used. Our goal is to prove, that for each $j \in [1, N]_{\mathbb{N}}$, there are at most two values $k \in [0, T-1]_{\mathbb{N}}$, where the coefficients $\alpha_{j,k}$

concentrate at, meaning that the sum of their squared magnitudes is large in comparison to the magnitude sums of the exponentially many other approximations, and, such that $2\pi k/t_0$ is a good approximation of λ_j. To get some intuition on this analysis, notice, that the value $\delta_{j,k}$ becomes very small if $2\pi k/t_0 \approx \lambda_j$. We try to carry this intuition over to the coefficients $\alpha_{j,k}$. The main result of this paragraph will be the following theorem.

Theorem 3.9. For the HHL phase estimation of an eigenvalue λ_j with $j \in [1, N]_{\mathbb{N}}$ arbitrary, but fixed, it holds, that

$$\sum_{\substack{k\in[0,T-1]_{\mathbb{N}} \\ |\delta_{j,k}|\geq 2\pi}} |\alpha_{j,k}|^2 < \frac{7}{10} \tag{3.3.10}$$

Let j, k as in Equation (3.3.8) be arbitrary, but fixed with $|\delta_{j,k}| \geq 2\pi$.

Observation 3.10. By definition and $\lambda_j \in \left[\frac{1}{\kappa}, 1\right]$, we have $|\delta_{j,k}| \leq \max(\{t_0, 2\pi(T-1)\})$. Since, due to the algorithm description, we assume a choice of t, s.t. $t_0 \leq 2\pi(T-1)$, as $T \geq t_0/2\pi + 1$, we get

$$2\pi \leq |\delta_{j,k}| \leq 2\pi(T-1)$$

We disagree with the assumption by the HHL authors, that $|\delta_{i,j}| \leq T/10$ [3, p. 11]. If one applies the algorithm on any unit matrix and chooses t_0 to be incredibly high, whilst using only few helper qubits, the bound will not hold. In the following, we present a detailled derivation of the alternative representation of the $\alpha_{j,k}$ values, which can be found in [3, p. 11].

Lemma 3.11. The following bounds hold for any $x \in \mathbb{R}_{\geq 0}$:

$$x - \frac{x^3}{6} \leq \sin(x) \leq x \tag{3.3.11}$$

With strict inequalities for $x \neq 0$.

The lower bound is not obvious. We present the proof of this rather elementary bound in Appendix A.

Lemma 3.12. The following statements hold.

(i) For $\delta_{j,k} \notin \{\pm\pi\}$, it holds, that

$$\alpha_{j,k} = -e^{i\frac{\delta_{j,k}}{2}\left(1-\frac{1}{T}\right)} \frac{\sqrt{2}\cos\left(\frac{\delta_{j,k}}{2}\right)}{T} \frac{\cos\left(\frac{\delta_{j,k}}{2T}\right)\sin\left(\frac{\pi}{2T}\right)}{\sin\left(\frac{\delta_{j,k}+\pi}{2T}\right)\sin\left(\frac{\delta_{j,k}-\pi}{2T}\right)} \tag{3.3.12}$$

(ii) The function

$$\xi\colon (-2\pi, 2\pi) \setminus \{\pm\pi\} \to \mathbb{R}, \delta \mapsto \frac{2}{T^2}\sin^2\left(\frac{\pi}{2T}\right) \frac{\cos^2\left(\frac{\delta}{2}\right)\cos^2\left(\frac{\delta}{2T}\right)}{\sin^2\left(\frac{\delta+\pi}{2T}\right)\sin^2\left(\frac{\delta-\pi}{2T}\right)} \tag{3.3.13}$$

can be continuously extended to $(-2\pi, 2\pi)$.

Proof. (i) We give thorough explanations to the following large computation.

$$\alpha_{j,k} = \frac{\sqrt{2}}{T}\sum_{\tau=0}^{T-1}\sin\left(\frac{\pi(\tau+\frac{1}{2})}{T}\right)\exp\left(\frac{i\tau}{T}\delta_{j,k}\right) \tag{3.3.14}$$

$$\overset{(1)}{=} \frac{1}{i\sqrt{2}T}\sum_{\tau=0}^{T-1}\exp\left(\frac{i\tau}{T}\delta_{j,k}\right)\left(\exp\left(i\frac{\pi(\tau+\frac{1}{2})}{T}\right)-\exp\left(-i\frac{\pi(\tau+\frac{1}{2})}{T}\right)\right) \tag{3.3.15}$$

$$\overset{(2)}{=} \frac{1}{i\sqrt{2}T}\left(\exp\left(\frac{i\pi}{2T}\right)\sum_{\tau=0}^{T-1}\exp\left(i\tau\frac{\delta_{j,k}+\pi}{T}\right)-\exp\left(-\frac{i\pi}{2T}\right)\sum_{\tau=0}^{T-1}\exp\left(i\tau\frac{\delta_{j,k}-\pi}{T}\right)\right) \tag{3.3.16}$$

$$\overset{(3)}{=} \frac{1}{i\sqrt{2}T}\left(\exp\left(\frac{i\pi}{2T}\right)\frac{1-e^{i(\delta_{j,k}+\pi)}}{1-e^{i\frac{\delta_{j,k}+\pi}{T}}}-\exp\left(-\frac{i\pi}{2T}\right)\frac{1-e^{i(\delta_{j,k}-\pi)}}{1-e^{i\frac{\delta_{j,k}-\pi}{T}}}\right) \tag{3.3.17}$$

$$\overset{(4)}{=} \frac{1+e^{i\delta_{j,k}}}{i\sqrt{2}T}\left(\frac{e^{-i\frac{\delta_{j,k}}{2T}}}{e^{-i\frac{\delta_{j,k}+\pi}{2T}}-e^{i\frac{\delta_{j,k}+\pi}{2T}}}-\frac{e^{-i\frac{\delta_{j,k}}{2T}}}{e^{-i\frac{\delta_{j,k}-\pi}{2T}}-e^{i\frac{\delta_{j,k}-\pi}{2T}}}\right) \tag{3.3.18}$$

$$\overset{(5)}{=} \frac{(1+e^{i\delta_{j,k}})e^{-i\frac{\delta_{j,k}}{2T}}}{i\sqrt{2}T}\left(\frac{1}{-2i\sin\left(\frac{\delta_{j,k}+\pi}{2T}\right)}-\frac{1}{-2i\sin\left(\frac{\delta_{j,k}-\pi}{2T}\right)}\right) \tag{3.3.19}$$

$$\overset{(6)}{=} e^{i\frac{\delta_{j,k}}{2}(1-\frac{1}{T})}\frac{\cos\left(\frac{\delta_{j,k}}{2}\right)}{\sqrt{2}T}\frac{\sin\left(\frac{\delta_{j,k}-\pi}{2T}\right)-\sin\left(\frac{\delta_{j,k}+\pi}{2T}\right)}{\sin\left(\frac{\delta_{j,k}+\pi}{2T}\right)\sin\left(\frac{\delta_{j,k}-\pi}{2T}\right)} \tag{3.3.20}$$

$$\overset{(7)}{=} -e^{i\frac{\delta_{j,k}}{2}(1-\frac{1}{T})}\frac{\sqrt{2}\cos\left(\frac{\delta_{j,k}}{2}\right)}{T}\frac{\cos\left(\frac{\delta_{j,k}}{2T}\right)\sin\left(\frac{\pi}{2T}\right)}{\sin\left(\frac{\delta_{j,k}+\pi}{2T}\right)\sin\left(\frac{\delta_{j,k}-\pi}{2T}\right)} \tag{3.3.21}$$

(1) Use the definition of the sine with the complex exponential function, see Definition B.2.

(2) Reorder the terms wrt. the dependency on τ.

(3) Use the geometric sum. With $\delta_{j,k} \notin \{\pm\pi\}$, it is assured, that we do not add up ones, as otherwise the geometric sum does not apply here in this form.

(4) Notice $e^{i(\delta_{j,k}+\pi)} = -e^{i\delta_{j,k}} = e^{i(\delta_{j,k}-\pi)}$ by definition. Thus, we first factor out $1+e^{i\delta_{j,k}}$. Expand the terms by $e^{-i\frac{\delta_{j,k}+\pi}{2T}}$ and $e^{-i\frac{\delta_{j,k}-\pi}{2T}}$, through which the factors $e^{\frac{i\pi}{2T}}$ and $e^{-\frac{i\pi}{2T}}$ get cancelled out.

(5) Factor out the numerators and use the definition of the sine via the complex exponential function, see Definition B.2, in the denominators.

(6) Factoring out $1/(-2i)$ from the sums yields a denominator of $2\sqrt{2}T$. Now, using the exponential form of the *cosine* function for once, we also obtain:

$$(1+e^{i\delta_{j,k}})e^{-i\frac{\delta_{j,k}}{2T}} = (e^{-i\frac{\delta_{j,k}}{2}}+e^{i\frac{\delta_{j,k}}{2}})e^{i\frac{\delta_{j,k}}{2}(1-\frac{1}{T})} = 2\cos\left(\frac{\delta_{j,k}}{2}\right)e^{i\frac{\delta_{j,k}}{2}(1-\frac{1}{T})} \tag{3.3.22}$$

At last, we expand the right terms. This fixes one calculation mistake of the original paper: The $\sqrt{2}$ is part of the denominator.

(7) We use the sine addition theorem, see Theorem B.4. With the asymmetry of the sine function, and the symmetry of the cosine function, this yields:

$$\sin\left(\frac{\delta_{j,k}-\pi}{2T}\right)-\sin\left(\frac{\delta_{j,k}+\pi}{2T}\right) = \cos\left(\frac{\delta_{j,k}}{2T}\right)\sin\left(\frac{-\pi}{2T}\right)-\cos\left(\frac{\delta_{j,k}}{2T}\right)\sin\left(\frac{\pi}{2T}\right) \tag{3.3.23}$$

$$= -2\cos\left(\frac{\delta_{j,k}}{2T}\right)\sin\left(\frac{\pi}{2T}\right) \tag{3.3.24}$$

(ii) Notice $\xi(-\delta) = \xi(\delta)$ due to $\sin^2\left(\frac{-\delta+\pi}{2T}\right)\sin^2\left(\frac{-\delta-\pi}{2T}\right) = \sin^2\left(\frac{\delta+\pi}{2T}\right)\sin^2\left(\frac{\delta-\pi}{2T}\right)$ and the axial symmetry of the cosine. It thus suffices to prove the existence of $\lim_{\delta\to\pi}\xi(\delta)$. We have

$$\lim_{\delta\to\pi}\frac{\cos\left(\frac{\delta}{2}\right)\cos\left(\frac{\delta}{2T}\right)}{\sin\left(\frac{\delta+\pi}{2T}\right)\sin\left(\frac{\delta-\pi}{2T}\right)} = \frac{\frac{\partial}{\partial\delta}\cos\left(\frac{\delta}{2}\right)\cos\left(\frac{\delta}{2T}\right)\big|_{\pi}}{\frac{1}{2T}\sin\left(\frac{\pi}{T}\right)} \tag{3.3.25}$$

using the rule of Bernoulli-L'Hospital [37, pp. 150-151] and

$$\frac{\partial}{\partial\delta}\sin\left(\frac{\delta+\pi}{2T}\right)\sin\left(\frac{\delta-\pi}{2T}\right)=\frac{1}{2T}\left(\cos\left(\frac{\delta+\pi}{2T}\right)\sin\left(\frac{\delta-\pi}{2T}\right)+\sin\left(\frac{\delta+\pi}{2T}\right)\cos\left(\frac{\delta-\pi}{2T}\right)\right) \tag{3.3.26}$$

$$=\frac{1}{2T}\sin\left(\frac{\delta}{T}\right) \tag{3.3.27}$$

using the product rule of differential calculus and Theorem B.4. Taking the continuity of $\delta\mapsto\delta^2$ into account, we obtain the statement.

This concludes the proof. ■

Part (ii) of this lemma will not be used, its importance lies in the fact, that the $\alpha_{j,k}$ values do not "blow up" for values $|\delta_{j,k}|$ near π. We now present a descriptive analytic proof for the bound concentration, i.e., that for j and k with $|\delta_{j,k}|<2\pi$, we have a good approximation of λ_j by $\frac{2\pi k}{t_0}$. It holds, that

$$|\alpha_{j,k}|\overset{(1)}{=}\frac{\sqrt{2}\left|\cos\left(\frac{\delta_{j,k}}{2}\right)\right|}{T}\frac{\left|\cos\left(\frac{\delta_{j,k}}{2T}\right)\right|\sin\left(\frac{\pi}{2T}\right)}{\sin\left(\frac{\delta_{j,k}+\pi}{2T}\right)\sin\left(\frac{\delta_{j,k}-\pi}{2T}\right)}\overset{(2)}{<}\frac{\pi}{\sqrt{2}T^2}\frac{1}{\sin\left(\frac{\delta_{j,k}+\pi}{2T}\right)\sin\left(\frac{\delta_{j,k}-\pi}{2T}\right)} \tag{3.3.28}$$

(1) Taking the complex magnitude respects products, $\left|-\exp\left(i\frac{\delta_{j,k}}{2}\left(1-\frac{1}{T}\right)\right)\right|=1$ and $\frac{\pi}{2T}\in\left(0,\frac{\pi}{64}\right)$, where the sine is positive. Furthermore, for $\delta_{j,k}>0$, we have $\frac{\delta_{j,k}+\pi}{2T},\frac{\delta_{j,k}-\pi}{2T}\in\left[\frac{\pi}{2T},\pi-\frac{\pi}{2T}\right]$, where the sine is also positive. We also have $\sin\left(\frac{-\delta_{j,k}+\pi}{2T}\right)\sin\left(\frac{-\delta_{j,k}-\pi}{2T}\right)=\sin\left(\frac{\delta_{j,k}-\pi}{2T}\right)\sin\left(\frac{\delta_{j,k}+\pi}{2T}\right)$, so we can leave out taking the magnitude again.

(2) Since $|\cos|\leq 1$ with strict inequality for arguments outside of $\pi\mathbb{Z}$, and Lemma 3.11.

We want to further study the result analytically.

Lemma 3.13. Define the auxiliary function

$$h\colon\mathbb{R}\setminus\{2\pi kT\pm\pi\mid k\in\mathbb{Z}\}\to\mathbb{R}_{>0},\delta\mapsto\frac{\pi}{\sqrt{2}T^2}\frac{1}{\sin\left(\frac{\delta+\pi}{2T}\right)\sin\left(\frac{\delta-\pi}{2T}\right)} \tag{3.3.29}$$

and let $h^{\pm}\coloneqq h|_{[\pm2\pi(T-1),\pm2\pi]\cup[\pm2\pi,\pm2\pi(T-1)]}$ each yielding h^+ and h^-. We have

(i) $h^-(-\delta)=h^+(\delta)$ for $\delta\in[2\pi,2\pi(T-1)]$.

(ii) h^+ is symmetric wrt. πT.

(iii) $h^+|_{[2\pi,\pi T]}$ strictly descends.

Proof. (i) We have proven this in (1) for Equation (3.3.28).

(ii) If $\delta\in[2\pi,\pi T]$, then $2\pi T-\delta\in[\pi T,2\pi(T-1)]$. Especially

$$\sin\left(\frac{(2\pi T-\delta)+\pi}{2T}\right)\sin\left(\frac{(2\pi T-\delta)-\pi}{2T}\right)=\sin\left(\pi-\frac{\delta-\pi}{2T}\right)\sin\left(\pi-\frac{\delta+\pi}{2T}\right) \tag{3.3.30}$$

$$=\sin\left(\frac{\delta-\pi}{2T}\right)\sin\left(\frac{\delta+\pi}{2T}\right) \tag{3.3.31}$$

(iii) As $\frac{\partial}{\partial\delta}\sin\left(\frac{\delta+\pi}{2T}\right)\sin\left(\frac{\delta-\pi}{2T}\right)=\frac{1}{2T}\sin\left(\frac{\delta}{T}\right)\geq 0$ in $[2\pi,\pi T]$, see Equation (A.0.7), the function stricly descends. Note, that the right bound does not matter by the definition of strict monotonicity.

■

Remark 3.14. The consequence of this lemma is, that we can reduce the calculation of h for any $\delta_{j,k}$ by mirroring the value at most twice, once around $x=0$ and once around $x=\pi T$. As the $\delta_{j,k}$ values are evenly spread on an interval of length $2\pi(T-2)$, we can further bound any sum over all $\alpha_{j,k}$ values by only considering the values of h in $[2\pi,\pi(T-1)]$. We shall use this thought in the proof of Theorem 3.9.

Lemma 3.15. Defining for $T \coloneqq 2^t$, $t \in \mathbb{N}_{\geq 5}$

$$l^{\uparrow} \colon [2\pi, \pi T] \to \mathbb{R}, \delta \mapsto \sin\left(\frac{\delta+\pi}{2T}\right)\sin\left(\frac{\delta-\pi}{2T}\right) \qquad l^{\downarrow} \colon [2\pi, \pi T] \to \mathbb{R}, \delta \mapsto \frac{c_1}{\pi^2}\frac{\delta^2}{T^2} \tag{3.3.32}$$

where $c_1 \coloneqq 0.9975 < \sin\left(\frac{\pi}{2} - \frac{\pi}{64}\right)$, we have $l^{\uparrow} > l^{\downarrow}$.

We leave out the technicalities of this lemma and prove it in the appendix, see Appendix A. Consider the illustration with a summary of the argument in Figure 12.

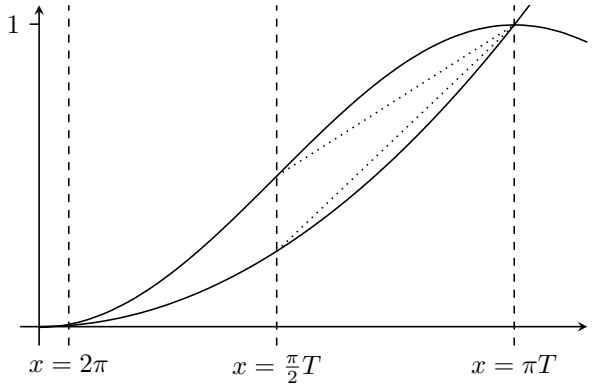


Figure 12: Graph of $l^{\uparrow}$ and $l^{\downarrow}$ für $t = 5$. The x-axis is scaled by $1/T$, the y-axis is scaled by 2 and the entire plot is scaled by 2. The vertical lines $x = 2\pi$, $x = \frac{\pi}{2}T$ and $x = \pi T$ are marked. In the interval $[2\pi, \pi T/2]$, $l^{\uparrow}$ grows faster than $l^{\downarrow}$, while being larger at the interval boundaries. In $[\pi T/2, \pi T]$, $l^{\uparrow}$ is convex, and larger at the boundary points, while $l^{\downarrow}$ is concave. The convexity and concavity argument is illustrated by the dotted lines. These facts conclude $l^{\uparrow} > l^{\downarrow}$. The rigorous formulation can be found in the appendix, as said.

Lemma 3.16. For $2\pi \leq \delta_{j,k} \leq \pi T$, we have

$$|\alpha_{j,k}| < \frac{22}{\delta_{j,k}^2} \tag{3.3.33}$$

Proof. Lemma 3.15 and Lemma 3.12 directly give us

$$|\alpha_{j,k}| < \frac{\pi}{\sqrt{2}T^2}\frac{\pi^2}{c_1}\frac{T^2}{\delta_{j,k}^2} < \frac{22}{\delta_{j,k}^2} \tag{3.3.34}$$

■

Remark 3.17. For a guarantee of a good approximation of the eigenvalues, Equation (A.0.12) shows, that at least five qubits are needed.

Now we are able to give the proof of the main theorem of this subsection.

Proof of Theorem 3.9. Due to the behavior of h in Lemma 3.13, we have

$$\sum_{\substack{k\in[0,T-1]_{\mathbb{N}} \\ |\delta_{j,k}|\geq 2\pi}} |\alpha_{j,k}|^2 \overset{(1)}{<} \sum_{k=1}^{T-1} h(2\pi k)^2 \overset{(2)}{<} 2\sum_{k=1}^{\infty} \frac{22^2}{16\pi^4 k^4} \overset{(3)}{=} 2 \cdot \frac{22^2}{16 \cdot 90} < \frac{7}{10} \tag{3.3.35}$$

(1) The values $\delta_{j,k}$ are positioned in distance 2π to each closest neighbor. Consider for each k, that $|\delta_{j,k}| \in [2\pi k', 2\pi(k'+1)]$ for some $k' \in \mathbb{Z}$. With the monotonicity behavior of h, as in Lemma 3.13, we thus have the upper bound using h^2 at either $2\pi k'$ or $2\pi(k'+1)$ for this value $|\alpha_{j,k}|^2$. To observe the statement for the sum, consider for a k with $\delta_{j,k} < 0$, that mirroring, i.e. taking $h^2(-\delta_{j,k})$ for the strict upper bound, and then mirroring at πT, gives the upper bound, and that no other element is then contained in the associated 2π-sized interval, in which $2\pi T + \delta_{j,k}$ lies, as then we have $\delta_{j,0} - \delta_{j,T-1} > 2\pi(T-1)$, which is a contradiction to the definition of the $\delta_{j,k}$ values.

(2) Following our previous considerations, we upper bound the values for $\delta_{j,k} \in [2\pi, \pi T]$ twice and let them tend to infinity for a constant upper bound.

(3) Using Theorem B.7.

■

Remark 3.18. With the last theorem proven, we will now use the notation $|\tilde{\lambda_k}\rangle \coloneqq |k\rangle$, $k \in [0, T-1]_{\mathbb{N}}$, following [3, p. 6], for the basis states, where $\tilde{\lambda_k} \coloneqq \frac{2\pi k}{t_0}$. Note that with this notation, we indicate that for some such k, $\tilde{\lambda_k}$ gives a good approximation for some eigenvalue λ_j, $j \in [1, N]_{\mathbb{N}}$.

The following theorem further elaborates the existence of a close approximation.

Theorem 3.19 (Existence of Eigenvalue Approximations)**.** The following statements hold for any fixed $j \in [1, N]_{\mathbb{N}}$.

(i) There is a $k \in [0, T-1]_{\mathbb{N}}$ with $|\delta_{j,k}| < 2\pi$.

(ii) For every $k \in [0, T-1]_{\mathbb{N}}$ with $|\delta_{j,k}| < 2\pi$, we have

$$|\tilde{\lambda_k} - \lambda_j| < \frac{1}{4\kappa} \tag{3.3.36}$$

and thus

$$\tilde{\lambda_k} \in \left(\frac{3}{4\kappa}, 1 + \frac{1}{4\kappa}\right) \tag{3.3.37}$$

Proof. We prove the statements in series.

(i) Fix j and consider under the condition $\delta_{j,k} \leq 0$

$$0 \leq -\delta_{j,k} = 2\pi k - \lambda_j t_0 < 2\pi \rightsquigarrow \frac{\lambda_j t_0}{2\pi} < k < \frac{\lambda_j t_0}{2\pi} + 1 \tag{3.3.38}$$

as $\frac{\lambda_j t_0}{2\pi} \in (0, T-1)$, we may thus choose $k \coloneqq \left\lfloor \frac{\lambda_j t_0}{2\pi} \right\rfloor$ or $k = 1$ for $\frac{\lambda_j t_0}{2\pi} < 1$.

(ii) From the assumption and the definition of $\delta_{j,k}$, we directly have

$$|\lambda_j - \tilde{\lambda_k}| = \frac{|\delta_{j,k}|}{t_0} < \frac{2\pi}{t_0} = \frac{\pi\varepsilon}{100\kappa} \in \left(0, \frac{1}{4\kappa}\right) \tag{3.3.39}$$

The second statement follows directly.

■

This proof is the main reason for our choice of ε. In the second next subsection, we will also see the reason for our choice of t_0.

Inversion of the Eigenvalue Approximations

With the eigenvalue approximations stored, we want to transfer them over into the amplitudes to obtain a form as in Corollary 1.12. The goal, as in Equation (3.2.2), is to obtain

$$\sum_{i=1}^{N} \frac{\beta_i}{\lambda_i} |v_i\rangle \tag{3.3.40}$$

in the input register. Rotating conditioned on a map of form $\lambda \mapsto \arcsin(C/\lambda)$, where $C \in (0, \lambda_{\min}(A)] \supseteq (0, \frac{1}{\kappa}]$ for normalization and $\lambda_{\min}(A)$ as in Definition 1.23, would suffice for this task, as we can verify by following along the calculations of the proof of Theorem 2.17. By measuring the helper qutrit and using amplitude amplification, we could obtain an approximation of the target state. The problem with this

approach stems from the case, when the eigenvalues are incredibly small. Errors in the phase estimation, which can be quite large simply due large $\delta_{j,k}$ values, see Figure 11, and by a poorly chosen t_0 value, can lead to poor result states. We need a more numerically stable procedure [12, p. 33].
The HHL authors have their own procedure $\mathcal{R}$, as mentioned in Algorithm 2. The filter functions are piecewise continuous functions, induced by concatenations of the sine, cosine and inversion, as well as constant functions. Especially, in $\left[\frac{1}{2\kappa}, \frac{1}{\kappa}\right)$, the sine and cosine functions have been concatenated with a linear transform, which transforms this interval into $\left[0, \frac{\pi}{2}\right)$ isomorphically. This helps us understand the filter functions more: We want an approximately continuous function f, that slowly descends for eigenvalues, which are not in the desired range, and a function g, which becomes large for bad eigenvalues to fend off unacceptibly small eigenvalues. For an illustration, consider Figure 13. We further need a qutrit and this specific choice of the functions for a working analysis of the error.

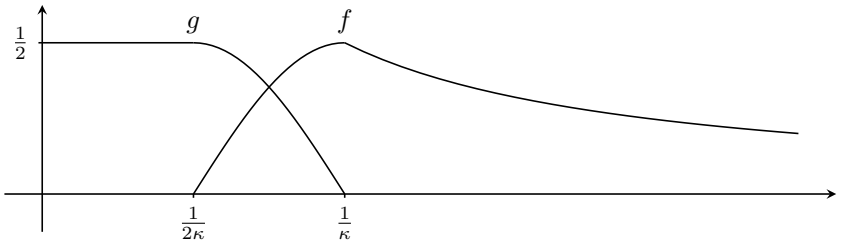


Figure 13: Sketch of the filter functions. Here an example for a matrix with eigenvalues $1, 4, 7, 10$ and thus $\kappa = 10$. The horizontal axis was scaled by 20, the vertical one by 2. One can very well see the rather sudden drop of g and the simultaneous entry of f in the interval $\left[\frac{1}{2\kappa}, \frac{1}{\kappa}\right]$.

The figure also demonstrates the effect of upper bounding κ, which is that tinier eigenvalues are permitted for inversion. The qutrit rotation technique described in Theorem 2.17 is also only applicable due to the following lemma.

Lemma 3.20. We have

$$f^2 + g^2 \leq \frac{1}{4} \tag{3.3.41}$$

with equality in $\left[0, \frac{1}{\kappa}\right]$.

Proof. Following the definitions in Equation (3.2.8), in $\left[0, \frac{1}{2\kappa}\right)$, $f^2 + g^2 = \frac{1}{4}$, in $\left[\frac{1}{2\kappa}, \frac{1}{\kappa}\right)$, we use Theorem B.4 to have $f^2 + g^2 = \frac{1}{4}$ and in $\left[\frac{1}{\kappa}, \infty\right)$, $f^2 + g^2 \leq \frac{1}{4}$ with $f^2\left(\frac{1}{\kappa}\right) = \frac{1}{4}$. ■

By application of $\mathcal{R}$, we obtain:

$$\sum_{j=1}^{N} \beta_j \sum_{k=0}^{T-1} \alpha_{j,k} \left|\tilde{\lambda_k}\right\rangle \left|v_j\right\rangle \left(\sqrt{1 - f^2(\tilde{\lambda_k}) - g^2(\tilde{\lambda_k})} \left|0\right\rangle + f(\tilde{\lambda_k}) \left|1\right\rangle + g(\tilde{\lambda_k}) \left|2\right\rangle\right) \tag{3.3.42}$$

As in Figure 8, we uncompute the first two registers. Keeping the concentration of the eigenvalue approximations in mind and following the calculations from before we gave the proof of Theorem 3.9, we apply $(\mathcal{T}^\dagger \otimes E_{N\cdot 3})(\mathrm{CHE}^\dagger_{T,N,A,t_0} \otimes E_3)(\mathrm{QFT}_T \otimes E_{N\cdot 3})$ after making an approximation, giving

$$\sum_{j=1}^{N} \beta_j \sum_{k=0}^{T-1} \alpha_{j,k} \left|\tilde{\lambda_k}\right\rangle \left|v_j\right\rangle \left(\sqrt{1 - f^2(\tilde{\lambda_k}) - g^2(\tilde{\lambda_k})} \left|0\right\rangle + f(\tilde{\lambda_k}) \left|1\right\rangle + g(\tilde{\lambda_k}) \left|2\right\rangle\right) \tag{3.3.43}$$

$$\approx \sum_{j=1}^{N} \beta_j \sum_{k=0}^{T-1} \alpha_{j,k} \left|\tilde{\lambda_k}\right\rangle \left|v_j\right\rangle \left(\sqrt{1 - f^2(\tilde{\lambda_j}) - g^2(\tilde{\lambda_j})} \left|0\right\rangle + f(\tilde{\lambda_j}) \left|1\right\rangle + g(\tilde{\lambda_j}) \left|2\right\rangle\right) \tag{3.3.44}$$

$$\mapsto \sum_{j=1}^{N} \beta_j \left|0\right\rangle \left|v_j\right\rangle \left(\sqrt{1 - f^2(\tilde{\lambda_j}) - g^2(\tilde{\lambda_j})} \left|0\right\rangle + f(\tilde{\lambda_j}) \left|1\right\rangle + g(\tilde{\lambda_j}) \left|2\right\rangle\right) \tag{3.3.45}$$

where $\tilde{\lambda_j}$ denotes the best approximation of λ_j for each j. Due to the possible entanglement of the ancilla qutrit with the clock register, we use Theorem 3.9 to enable this approximation. We now apply amplitude amplification with the procedure so far and χ as defined in Algorithm 2 to perform a measurement of a 1 in the ancilla qutrit. So for one measurement, assuming, that $\tilde{\lambda_j} \in \left[\frac{1}{\kappa}, \infty\right)$ for all $j \in [1, N]_{\mathbb{N}}$, we obtain the state

$$\frac{\sum_{j=1}^{N} \beta_j \frac{1}{2\kappa\tilde{\lambda_j}} |v_j\rangle}{\sqrt{\sum_{j=1}^{N} |\beta_j|^2 \left|\frac{1}{2\kappa\tilde{\lambda_k}}\right|^2}} \approx \frac{\sum_{j=1}^{N} \frac{\beta_j}{\lambda_j} |v_j\rangle}{\sqrt{\sum_{j=1}^{N} \left|\frac{\beta_j}{\lambda_j}\right|^2}} \text{ with probability } \sum_{j=1}^{N} |\beta_j|^2 \left|\frac{1}{2\kappa\tilde{\lambda_j}}\right|^2 \tag{3.3.46}$$

in the input register. We now make the latter parts of the argument precise.

Choosing the Evolution Time

For the phase estimation procedure to be successful, a well-chosen evolution time t_0 is required. We follow along the error analysis of Harrow et al. [3, pp. 7-10] for the following paragraph.
First, we need to generalize a small definition from real analysis. Recall the analytical concept of *Lipschitz-continuity*.

Definition 3.21. Let $m, n \in \mathbb{N}_{\geq 1}$ and $D \subseteq \mathbb{R}^m$. A function $f\colon D \to \mathbb{R}^n$ is called *Lipschitz-continuous* with *Lipschitz-constant* $C \in \mathbb{R}_{\geq 0}$, if for any $\lambda, \lambda' \in D$, it holds that:

$$\|f(\lambda) - f(\lambda')\| \leq C\|\lambda - \lambda'\| \tag{3.3.47}$$

We also call f *C-Lipschitz*.

Theorem 3.22. Let $U \subseteq \mathbb{R}$ be an open, convex subset and $f\colon U \to \mathbb{R}^n$ be continuous differentiable. f is Lipschitz-continuous, iff $f' = (f_1', ..., f_n')$ is bounded. Furthermore, if for a $C \in \mathbb{R}_{\geq 0}$, we have $\|f'\| \leq C$, then f is C-Lipschitz.

Proof. Let $\lambda, \lambda' \in U$ with $\lambda \neq \lambda'$.
($\Rightarrow$) Let $C \in \mathbb{R}_{>0}$ be the Lipschitz-constant. We get:

$$\left\| \frac{f(\lambda) - f(\lambda')}{\lambda - \lambda'} \right\| \leq C \tag{3.3.48}$$

Taking the limit for $\lambda' \to \lambda$ yields $\|f'(\lambda)\| \leq C$.
($\Leftarrow$) Let $C \in \mathbb{R}_{>0}$, s.t. $\|f'\| \leq C$. Due to convexity, $\{\lambda + t(\lambda' - \lambda) \mid t \in [0, 1]\} \subset U$. Using the multi-dimensional mean value theorem [8, p. 84] and the monotonicity of the integral, we obtain:

$$\|f(\lambda') - f(\lambda)\| = \left\| \int_0^1 f'(\lambda + t(\lambda' - \lambda))\, dt \right\| |\lambda' - \lambda| \leq C|\lambda' - \lambda| \tag{3.3.49}$$

Which is the statement. By that, we also have, that f is C-Lipschitz. ■

Remark 3.23. Note that a Lipschitz-constant C is an upper-bound on the derivative, and an upper-bound on the derivative C is a Lipschitz-constant.

Remark 3.24. We possibly could relax the assumptions on convexity, continuous differentiability and allow more arguments, but this suffices for our use case.

From the qutrit rotation performed in Section 3.3, we define the following map:

$$|h(\cdot)\rangle : \mathbb{R} \to \mathbb{C}^3, \lambda \mapsto |h(\lambda)\rangle \coloneqq \sqrt{1 - f^2(\lambda) - g^2(\lambda)}\, |0\rangle + f(\lambda)\, |1\rangle + g(\lambda)\, |2\rangle \tag{3.3.50}$$

We now prove three lemmata.

Lemma 3.25. The map $|h(\cdot)\rangle$ is $\frac{\pi}{2}\kappa$-Lipschitz.

This proof is a rewrite of the proof at [3, p. 7].

Proof. The statement is clear in $\left[0, \frac{1}{2\kappa}\right)$, as the filter functions are constant there, meaning that $|h(\lambda)\rangle - |h(\lambda')\rangle = 0$ for any $\lambda, \lambda' \in \left[0, \frac{1}{2\kappa}\right)$, and the statement follows from the definition of norms. $|h(\cdot)\rangle$ is continuous and differentiable in $\mathbb{R} \setminus \left\{\frac{1}{2\kappa}, \frac{1}{\kappa}\right\}$, due to the components. Due to Theorem 3.22, we may bound the derivatives in each subinterval. In $\left[\frac{1}{2\kappa}, \frac{1}{\kappa}\right)$, we have, using Theorem B.3,

$$\frac{\partial}{\partial\lambda}|h(\lambda)\rangle = \frac{\partial}{\partial\lambda}\left(\frac{1}{2}\sin\left(\frac{\pi}{2}\frac{\lambda-\frac{1}{2\kappa}}{\frac{1}{\kappa}-\frac{1}{2\kappa}}\right)|1\rangle + \frac{1}{2}\cos\left(\frac{\pi}{2}\frac{\lambda-\frac{1}{2\kappa}}{\frac{1}{\kappa}-\frac{1}{2\kappa}}\right)|2\rangle\right) \tag{3.3.51}$$

$$= \frac{1}{2}\frac{\pi}{2}\frac{1}{\frac{1}{\kappa}-\frac{1}{2\kappa}}\left(\cos\left(\frac{\pi}{2}\frac{\lambda-\frac{1}{2\kappa}}{\frac{1}{\kappa}-\frac{1}{2\kappa}}\right)|1\rangle - \sin\left(\frac{\pi}{2}\frac{\lambda-\frac{1}{2\kappa}}{\frac{1}{\kappa}-\frac{1}{2\kappa}}\right)|2\rangle\right) \tag{3.3.52}$$

Taking the norm, we get $\left\|\frac{\partial}{\partial\lambda}|h(\lambda)\rangle\right\| = \frac{\pi}{2}\kappa$. We now look at $\left[\frac{1}{\kappa}, \infty\right)$. We get:

$$\frac{\partial}{\partial\lambda}|h(\lambda)\rangle = \frac{\partial}{\partial\lambda}\left(\sqrt{1-\frac{1}{4\kappa^2\lambda^2}}|0\rangle + \frac{1}{2\kappa\lambda}|1\rangle\right) = \frac{1}{2\kappa^2\lambda^3}\frac{1}{2}\frac{1}{\sqrt{1-1/(4\kappa^2\lambda^2)}}|0\rangle - \frac{1}{2\kappa\lambda^2}|1\rangle \tag{3.3.53}$$

We calculate the squared norm to receive

$$\left\|\frac{\partial}{\partial\lambda}|h(\lambda)\rangle\right\|^2 = \frac{1}{4\kappa^4\lambda^6}\frac{1}{4}\frac{1}{1-1/(4\kappa^2\lambda^2)} + \frac{1}{4\kappa^2\lambda^4} = \frac{1}{4\kappa^2\lambda^4}\left(\frac{1}{4\kappa^2\lambda^2-1}+1\right) \overset{(1)}{\leq} \frac{\kappa^2}{4}\left(\frac{4}{3}\right) = \frac{\kappa^2}{3} \tag{3.3.54}$$

(1) We use $\frac{1}{\kappa} \leq \lambda$.

The statement is thus true. ■

The following lemma, slightly adjusted from [3, pp. 9-10], gives a more specialized Lipschitz-type condition. Denote $f_j \coloneqq f(\lambda_j)$ and $\tilde{f}_k \coloneqq f(\tilde{\lambda_k})$, and analogously $g_j \coloneqq g(\lambda_j)$, $\tilde{g}_k \coloneqq g(\tilde{\lambda_k})$ for any $j \in [1, N]_{\mathbb{N}}$, $k \in [0, T-1]_{\mathbb{N}}$.

Lemma 3.26. It holds, that

$$(\tilde{f}_k - f_j)^2 + (\tilde{g}_k - g_j)^2 \leq \pi^2\frac{\kappa^2}{t_0^2}\delta_{j,k}^2(f_j^2+g_j^2) \tag{3.3.55}$$

Proof. We perform four case distinctions.

I. First, consider the case, where $\lambda_j \geq \frac{1}{\kappa}$ and $\tilde{\lambda_k} \geq \frac{1}{\kappa}$ as well, then $g_j = \tilde{g}_k = 0$ and we have using the definitions, $\frac{\delta_{j,k}}{t_0} = \frac{\lambda_j t_0}{t_0} - \frac{2\pi k}{t_0} = \lambda_j - \tilde{\lambda_k}$ and the assumption:

$$\tilde{f}_k - f_j = \frac{1}{2\kappa}\frac{\lambda_j - \tilde{\lambda_k}}{\tilde{\lambda_k}\lambda_j} \leq \frac{1}{2}\frac{\delta_{j,k}}{t_0}\frac{1}{\lambda_j} < \pi\frac{\kappa}{t_0}\delta_{j,k}\frac{1}{2\kappa\lambda_j} = \pi\frac{\kappa}{t_0}\delta_{j,k}f_j \tag{3.3.56}$$

Squaring both sides gives the statement.

II. Now consider the case, where again $\lambda_j \geq \frac{1}{\kappa}$ and now $\tilde{\lambda_k} \in \left[0, \frac{1}{2\kappa}\right)$. Then $\tilde{f}_k = 0$, $\tilde{g}_k = \frac{1}{2}$ and the claim is thus

$$f_j^2 + \frac{1}{4} \leq \pi^2\frac{\kappa^2}{t_0^2}\delta_{j,k}^2 f_j^2 \tag{3.3.57}$$

We first have

$$\frac{\pi^2}{2}\frac{\kappa^2}{t_0^2}\delta_{j,k}^2 f_j^2 = \frac{\pi^2}{8}\left(\frac{\lambda_j - \tilde{\lambda_k}}{\lambda_j}\right)^2 > \frac{\pi^2}{8}\frac{1}{4\kappa^2\lambda_j^2} > f_j^2 \tag{3.3.58}$$

but we also have

$$\frac{\pi^2}{8}\left(\frac{\lambda_j - \tilde{\lambda_k}}{\lambda_j}\right)^2 > \frac{\pi^2}{8}\left(1-\frac{1}{2\kappa\lambda_j}\right)^2 \geq \frac{\pi^2}{32} > \frac{1}{4} \tag{3.3.59}$$

due to $\inf_{\lambda\in\left[0,\frac{1}{2\kappa}\right)}(\lambda_j-\lambda)^2 = \left(\lambda_j - \frac{1}{2\kappa}\right)^2$. Adding both inequalities together gives the statement.

III. In the case of $\lambda_j \geq \frac{1}{\kappa}$ and $\tilde{\lambda_k} \in \left[\frac{1}{2\kappa}, \frac{1}{\kappa}\right)$, the claim becomes via Theorem B.3

$$(\tilde{f}_k - f_j)^2 + \tilde{g}_k^2 = \frac{1}{4\kappa^2\lambda_j^2} - \frac{1}{2\kappa\lambda_j}\sin\left(\frac{\pi}{2}(2\kappa\tilde{\lambda}_k - 1)\right) + \frac{1}{4} \leq \frac{\pi^2}{4}\left(\frac{\tilde{\lambda}_k - \lambda_j}{\lambda_j}\right)^2 = \pi^2\frac{\kappa^2}{t_0^2}\delta_{j,k}^2 f_j^2 \tag{3.3.60}$$

which is equivalent to

$$\kappa^2\lambda_j^2 + 2\kappa\lambda_j\cos\left(\pi\kappa\tilde{\lambda_k}\right) + 1 \leq \pi^2\kappa^2(\lambda_j - \tilde{\lambda_k})^2 \tag{3.3.61}$$

after division by f_j^2 on both sides and due to $\sin\left(x - \frac{\pi}{2}\right) = -\cos(x)$ for all $x \in \mathbb{R}$. Now fix λ_j. For $\tilde{\lambda_k} = \frac{1}{\kappa}$, which we may insert due to the continuity of f, the statement is

$$(\kappa\lambda_j - 1)^2 \leq \pi^2(\kappa\lambda_j - 1)^2 \tag{3.3.62}$$

which is true. Letting $\tilde{\lambda_k}$ be loose, we show, that the left hand side monotonically decreases slower than the right hand side, from which we conclude the inequality, as otherwise the right hand side would have already surpassed the left hand side when reaching $\frac{1}{\kappa}$. Applying $\frac{\partial}{\partial\tilde{\lambda_k}}$ on Equation (3.3.61) and dividing by $2\pi\kappa^2$ gives the condition

$$0 \geq -\lambda_j\sin\left(\pi\kappa\tilde{\lambda_k}\right) \geq \pi(\tilde{\lambda_k} - \lambda_j) \tag{3.3.63}$$

Both the left and right hand side in Equation (3.3.61) are thus montonically decreasing. The left hand side becomes 0 at $\frac{1}{\kappa}$. Align the associated tangent, which is

$$\left[\frac{1}{2\kappa}, \frac{1}{\kappa}\right] \to \mathbb{R}, \lambda \mapsto \pi\kappa\lambda_j\left(\lambda - \frac{1}{\kappa}\right) \tag{3.3.64}$$

Due to sine reaching its highest growth at $\frac{1}{\kappa}$, this tangent is a lower bound of the left hand side. So we have

$$-\lambda_j\sin\left(\pi\kappa\tilde{\lambda_k}\right) \geq \pi\kappa\lambda_j\left(\tilde{\lambda_k} - \frac{1}{\kappa}\right) = \pi\kappa\lambda_j\tilde{\lambda_k} - \pi\lambda_j \geq \pi(\tilde{\lambda_k} - \lambda_j) \tag{3.3.65}$$

due to $\kappa\lambda_j \geq 1$.

IV. Now consider the case, where $\lambda_j < 1/\kappa$. Then, by Lemma 3.25 and the definition of $|h(\cdot)\rangle$, $\frac{\delta_{j,k}}{t_0} = \lambda_j - \tilde{\lambda_k}$, as well as Lemma 3.20, we have

$$(\tilde{f}_k - f_j)^2 + (\tilde{g}_k - g_j)^2 \leq \| \, |h(\tilde{\lambda_k})\rangle - |h(\lambda_j)\rangle \, \|^2 \leq \frac{\pi^2}{4}\kappa^2(\tilde{\lambda_k} - \lambda_j)^2 = \pi^2\frac{\kappa^2}{t_0^2}\delta_{j,k}^2(f_j^2 + g_j^2) \tag{3.3.66}$$

This concludes the proof. ■

Lemma 3.27. Let $m, n \in \mathbb{N}_{\geq 1}$, $|\chi\rangle \in \mathbb{C}^m$ and $|\varphi\rangle, |\psi\rangle \in \mathbb{C}^n$ with $\||\chi\rangle\| = \||\varphi\rangle\| = \||\psi\rangle\| = 1$. Then, we have

$$\langle|\chi\rangle \otimes |\varphi\rangle||\chi\rangle \otimes |\psi\rangle\rangle = \langle\varphi|\psi\rangle \tag{3.3.67}$$

Proof. It holds, that

$$\langle|\chi\rangle\,|\varphi\rangle||\chi\rangle\,|\psi\rangle\rangle = \left\langle \begin{pmatrix} \chi_1\,|\varphi\rangle \\ \dots \\ \chi_m\,|\varphi\rangle \end{pmatrix} \middle| \begin{pmatrix} \chi_1\,|\psi\rangle \\ \dots \\ \chi_m\,|\psi\rangle \end{pmatrix} \right\rangle = \sum_{k=1}^{m}|\chi_i|^2\sum_{j=1}^{n}\varphi_j\psi_j^* = \sum_{j=1}^{n}\varphi_j\psi_j^* = \langle\varphi|\psi\rangle \tag{3.3.68}$$

■

Lemma 3.28. For arbitrary $p, \tilde{p} \in \mathbb{R}_{>0}$, we have:

$$\frac{\sqrt{p}}{\sqrt{\tilde{p}}} \geq 1 - \frac{1}{2}\frac{\tilde{p}-p}{p}$$

Proof. Fix p and introduce

$$l\colon (0,1] \to \mathbb{R}, p' \mapsto \frac{\sqrt{p}}{\sqrt{p'}} \tag{3.3.69}$$

and expand l into the first two terms of its Taylor series around p with the Lagrangian remainder term [38, p. 284], meaning that there is a $\xi \in [\tilde{p}, p] \cup [p, \tilde{p}]$ with

$$l(\tilde{p}) = \frac{l^{(0)}(p)}{0!}(\tilde{p}-p)^0 + \frac{l^{(1)}(p)}{1!}(\tilde{p}-p)^1 + \frac{l^{(2)}(\xi)}{2!}(\tilde{p}-p)^2 = 1 - \frac{1}{2}\frac{\tilde{p}-p}{p} + \frac{1}{2}\cdot\frac{3}{4}\frac{\sqrt{p}}{\sqrt{\xi^5}}(\tilde{p}-p)^2 \tag{3.3.70}$$

The last summand is positive, yielding the claim. ■

Now to the main theorem of this paragraph. Remember our assumption, that all subprocedures work without error. The only source of error comes from the phase estimation performed by the gates $\mathcal{T}$, CHE_{T,N,A,t_0} and $\mathrm{QFT}_T^\dagger$, after $|b\rangle$ has been initialized. Let

$$\tilde{P} \coloneqq (\mathcal{T} \otimes E_{N\cdot 3})(\mathrm{CHE}_{T,N,A,t_0} \otimes E_3)(\mathrm{QFT}_T^\dagger \otimes E_{N\cdot 3}) \tag{3.3.71}$$

and let P denote the version of $\tilde{P}$, which approximates the eigenvalues without error. Let U then denote the perfect HHL algorithm before the qutrit measurement using P with the result $|\varphi\rangle$ and let $\tilde{U}$ denote the imperfect algorithm using $\tilde{P}$ with $|\tilde{\varphi}\rangle$ being its result. Thus

$$U = P^\dagger \mathcal{R} P \qquad \tilde{U} = \tilde{P}^\dagger \mathcal{R} \tilde{P} \tag{3.3.72}$$

$$|\varphi\rangle = U|b\rangle \qquad |\tilde{\varphi}\rangle = \tilde{U}|b\rangle \tag{3.3.73}$$

The following main result from [3, pp. 7-10] gives the dependence of the overall algorithm error ε of $\tilde{U}$ on t_0, where exact bounds have been computed here.

Theorem 3.29 (Evolution Time for a Desired Error Cap)**.** The following statements hold.

(i) For the operator distance of the unitaries U and $\tilde{U}$, it holds, that:

$$\|U - \tilde{U}\| < 17\frac{\kappa}{t_0} \tag{3.3.74}$$

(ii) Suppose we measure $|\varphi\rangle$ and $|\tilde{\varphi}\rangle$ wrt. the observable $\{\mathrm{span}(\mathcal{B}_0'), \mathrm{span}(\mathcal{B}_1')\}$ with $\mathcal{B}_0' \coloneqq \{|x\rangle|y\rangle|0\rangle \mid (x,y) \in [0,T-1]_\mathbb{N} \times [0,N-1]_\mathbb{N}\}$, $\mathcal{B}_1' \coloneqq \{|x\rangle|y\rangle|1\rangle, |x\rangle|y\rangle|2\rangle \mid (x,y) \in [0,T-1]_\mathbb{N} \times [0,N-1]_\mathbb{N}\}$ and obtain the index 1, meaning that no zero was measured in the qutrit. Then

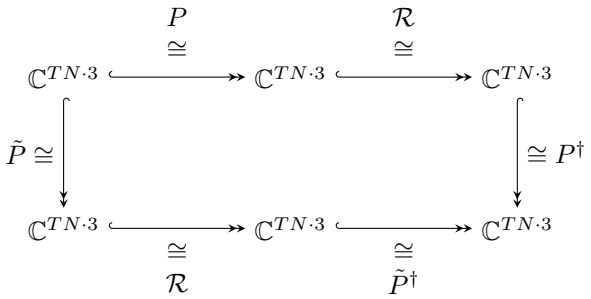
$$\||x'\rangle - |\tilde{x}'\rangle\| < 200\frac{\kappa}{t_0} \tag{3.3.75}$$

for the results $|x'\rangle, |\tilde{x}'\rangle \in \mathbb{C}^N$.

(iii) If, as assumed, A is well conditioned and thus all eigenvalues are inside of $\left[\frac{1}{\kappa}, 1\right]$, then after the final measurement, we have for the resulting states $|x\rangle, |\tilde{x}\rangle \in \mathbb{C}^N$ of the HHL algorithm

$$\||x\rangle - |\tilde{x}\rangle\| < 200\frac{\kappa}{t_0} \tag{3.3.76}$$

Note that for the first two statements, we do not require $\lambda_j \in \left[\frac{1}{\kappa}, 1\right]$ for all j, but for the last one. One way of illustrating the statement, is that, we *hope*, that the following diagram commutes:



Proof. We prove the statements in the order given.

(i) The goal is to bound the term $\|U|b\rangle - \tilde{U}|b\rangle\|$ for an arbitrary, but fixed $|b\rangle \in \mathbb{C}^N$, as that suffices for a bound on $\|U - \tilde{U}\|$, see Theorem 2.12. Writing out $|\varphi\rangle$ and $|\tilde{\varphi}\rangle$, we have

$$|\varphi\rangle = U|b\rangle = \sum_{j=1}^{N} \beta_j |0\rangle |v_j\rangle |h(\lambda_j)\rangle \qquad |\tilde{\varphi}\rangle = \tilde{U}|b\rangle = \tilde{P}^\dagger \sum_{j=1}^{N} \beta_j \sum_{k=0}^{T-1} \alpha_{j,k} |k\rangle |v_j\rangle \ |h(\tilde{\lambda_k})\rangle \tag{3.3.77}$$

Due to Theorem 2.11, it suffices to bound $\mathrm{Re}(\langle\varphi|\tilde{\varphi}\rangle)$ from below. Notice, that $\tilde{P}$ is unitary, thus isometric due to Theorem 1.5. This gives us:

$$\langle\varphi|\tilde{\varphi}\rangle = \langle\tilde{P}\varphi|\tilde{P}\tilde{\varphi}\rangle = \left\langle \sum_{j=1}^{N} \beta_j \sum_{k=0}^{T-1} \alpha_{j,k} |k\rangle |v_j\rangle |h(\lambda_j)\rangle \middle| \sum_{j=1}^{N} \beta_j \sum_{k=0}^{T-1} \alpha_{j,k} |k\rangle |v_j\rangle \ |h(\tilde{\lambda_k})\rangle \right\rangle \tag{3.3.78}$$

$$= \sum_{j=1}^{N} \sum_{k=0}^{T-1} |\beta_j \alpha_{j,k}|^2 \langle |k\rangle |v_j\rangle |h(\lambda_j)\rangle | |k\rangle |v_j\rangle \ |h(\tilde{\lambda_k})\rangle\rangle \tag{3.3.79}$$

$$\overset{(1)}{=} \sum_{j=1}^{N} \sum_{k=0}^{T-1} |\beta_j \alpha_{j,k}|^2 \langle h(\lambda_j)|h(\tilde{\lambda_k})\rangle \tag{3.3.80}$$

(1) We use Lemma 3.27.

From that, we have

$$\mathrm{Re}(\langle\varphi|\tilde{\varphi}\rangle) = \sum_{j=1}^{N} \sum_{k=0}^{T-1} |\beta_j \alpha_{j,k}|^2 \,\mathrm{Re}(\,\langle h(\lambda_j)|h(\tilde{\lambda_k})\rangle) \tag{3.3.81}$$

Using Lemma 3.25 and Theorem 2.11, we can observe:

$$\| |h(\lambda_j)\rangle - \ |h(\tilde{\lambda_k})\rangle\| = \sqrt{2(1 - \mathrm{Re}(\,\langle h(\lambda_j)|h(\tilde{\lambda_k})\rangle))} \leq \frac{\pi}{2}\kappa|\lambda_j - \tilde{\lambda_k}| \overset{(1)}{=} \frac{\pi}{2}\kappa \left|\frac{\delta_{j,k}}{t_0}\right| \tag{3.3.82}$$

$$\rightsquigarrow \mathrm{Re}(\,\langle h(\lambda_j)|h(\tilde{\lambda_k})\rangle) \geq 1 - \frac{\pi^2\kappa^2\delta_{j,k}^2}{8t_0^2} \tag{3.3.83}$$

(1) Since $\tilde{\lambda_k} = \frac{2\pi k}{t_0}$ by definition and thus $t_0(\lambda_j - \tilde{\lambda_k}) = \lambda_j t_0 - 2\pi k = \delta_{j,k}$.

We have

$$\|U|b\rangle - \tilde{U}|b\rangle\|^2 = 2(1-\mathrm{Re}(\langle\varphi|\tilde{\varphi}\rangle)) \leq 2\left(1-\left(\sum_{j=1}^{N}\sum_{k=0}^{T-1}|\beta_j\alpha_{j,k}|^2\left(1-\frac{\pi^2\kappa^2\delta_{j,k}^2}{8t_0^2}\right)\right)\right) \tag{3.3.84}$$

$$= 2\sum_{j=1}^{N}|\beta_j|^2\sum_{k=0}^{T-1}|\alpha_{j,k}|^2\frac{\pi^2\kappa^2\delta_{j,k}^2}{8t_0^2} = 2\sum_{j=1}^{N}|\beta_j|^2\left(\sum_{\substack{k\in\mathbb{N}_{\leq T-1}\\|\delta_{j,k}|<2\pi}}|\alpha_{j,k}|^2\delta_{j,k}^2+\sum_{\substack{k\in\mathbb{N}_{\leq T-1}\\|\delta_{j,k}|\geq 2\pi}}|\alpha_{j,k}|^2\delta_{j,k}^2\right)\frac{\pi^2}{8}\frac{\kappa^2}{t_0^2} \tag{3.3.85}$$

$$\overset{(1)}{<} 2\sum_{j=1}^{N}|\beta_j|^2\left(8\pi^2+\sum_{\substack{k\in\mathbb{N}_{\leq T-1}\\|\delta_{j,k}|\geq 2\pi}}\frac{22^2}{\delta_{j,k}^2}\right)\frac{\pi^2}{8}\frac{\kappa^2}{t_0^2} \overset{(2)}{<} 2\sum_{j=1}^{N}|\beta_j|^2\left(8\pi^2+22^2\cdot 2\sum_{k=1}^{\infty}\frac{1}{4\pi^2k^2}\right)\frac{\pi^2}{8}\frac{\kappa^2}{t_0^2} \tag{3.3.86}$$

$$\overset{(3)}{=} 2\left(8\pi^2+\frac{2\cdot 11^2\cdot\pi^2}{90}\right)\frac{\pi^2}{8}\frac{\kappa^2}{t_0^2} < 261\frac{\kappa^2}{t_0^2} \tag{3.3.87}$$

(1) We upper-bound using $|\delta_{j,k}| < 2\pi$ directly on the left summation. We further use Observation 3.8, where the summation is over at most two values. On the right sum, we use Lemma 3.16 directly.

(2) In the right term, use the fact, that the $\delta_{j,k}$ values differ by an integer multiple of 2π each to upper bound the summation term by the series $\sum_{k=1}^{\infty}\frac{1}{4\pi^2k^2}$. After obtaining the bound, which is only dependent on κ^2/t_0^2, we use $\sum_{j=1}^{N}|\beta_j|^2 = 1$.

(3) We use Theorem B.7. A consequence of this calculation is $\sum_{j=1}^{N}|\beta_j|^2\sum_{k=0}^{T-1}|\alpha_{j,k}|^2\delta_{j,k}^2 < 106$.

Since $|b\rangle$ was chosen arbitrarily and $\sqrt{261} < 17$, this concludes the proof.

(ii) We use the notation from Lemma 3.26. With the operator $\tilde{P}$ from above, we have:

$$|x'\rangle = \frac{\sum_{j=1}^{N}\beta_j|0\rangle|v_j\rangle(f_j|1\rangle+g_j|2\rangle)}{\sqrt{p'}} \text{ with } p' \coloneqq \sum_{j=1}^{N}|\beta_j|^2(f_j^2+g_j^2) \tag{3.3.88}$$

$$|\tilde{x}'\rangle = \frac{\tilde{P}^\dagger\sum_{j=1}^{N}\beta_j\sum_{k=0}^{T-1}\alpha_{j,k}|k\rangle|v_j\rangle(\tilde{f}_k|1\rangle+\tilde{g}_k|2\rangle)}{\sqrt{\tilde{p}'}} \text{ with } \tilde{p}' \coloneqq \sum_{j=1}^{N}\sum_{k=0}^{T-1}|\beta_j\alpha_{j,k}|^2(\tilde{f}_k^2+\tilde{g}_k^2) \tag{3.3.89}$$

Notice $p', \tilde{p}' \neq 0$. The maps $j \mapsto f_j^2+g_j^2$ and $(j,k)\mapsto\tilde{f}_k^2+\tilde{g}_k^2$ may be interpreted as random variables in this context, where the associated probabilities are given by $|\beta_j|^2$, and $|\beta_j\alpha_{j,k}|^2$ respectively. We thus have $p' = \mathrm{E}[f_j^2+g_j^2]$ and $\tilde{p}' = \mathrm{E}[\tilde{f}_k^2+\tilde{g}_k^2]$, where we omit the usual introduction of a formal random variable.

We now proceed with the notation from Lemma 3.26. With Lemma 3.27, it holds, that

$$\langle x'|\tilde{x}'\rangle = \langle\tilde{P}x'|\tilde{P}\tilde{x}'\rangle = \frac{\sum_{j=1}^{N}|\beta_j|^2\sum_{k=0}^{T-1}|\alpha_{j,k}|^2\langle f_j|1\rangle+g_j|2\rangle|\tilde{f}_k|1\rangle+\tilde{g}_k|2\rangle\rangle}{\sqrt{p'\tilde{p}'}} = \frac{\mathrm{E}[f_j\tilde{f}_k+g_j\tilde{g}_k]}{\sqrt{p'\tilde{p}'}} \tag{3.3.90}$$

Furthermore, with the linearity of the expectation value [39, p. 21], we have

$$\frac{\mathrm{E}[f_j\tilde{f}_k+g_j\tilde{g}_k]}{\sqrt{p'\tilde{p}'}} = \frac{\mathrm{E}[f_j^2+g_j^2]+\mathrm{E}[(\tilde{f}_k-f_j)f_j+(\tilde{g}_k-g_j)g_j]}{\sqrt{p'\tilde{p}'}} \tag{3.3.91}$$

$$= \frac{1+\mathrm{E}[(\tilde{f}_k-f_j)f_j+(\tilde{g}_k-g_j)g_j]/p'}{\sqrt{1+\frac{\tilde{p}'-p'}{p'}}} \tag{3.3.92}$$

$$\overset{(1)}{\geq}\left(1+\frac{\mathrm{E}[(\tilde{f}_k-f_j)f_j+(\tilde{g}_k-g_j)g_j]}{p'}\right)\left(1-\frac{1}{2}\frac{\tilde{p}'-p'}{p'}\right) \tag{3.3.93}$$

$$\overset{(2)}{=} 1-\frac{\mathrm{E}[(\tilde{f}_k-f_j)^2+(\tilde{g}_k-g_j)^2]}{2p'}-\frac{\mathrm{E}[(\tilde{f}_k-f_j)f_j+(\tilde{g}_k-g_j)g_j]}{p'}\frac{\tilde{p}'-p'}{2p'} \tag{3.3.94}$$

(1) By using Lemma 3.28 on $\frac{1}{\sqrt{1+\frac{\tilde{p}'-p'}{p'}}}$.

(2) Using the linearity of expectation and $\tilde{f}_k^2 - f_j^2 = (\tilde{f}_k - f_j)(\tilde{f}_k + f_j) = (\tilde{f}_k - f_j)(\tilde{f}_k - f_j + 2f_j)$, with the same statement for $\tilde{g}_k$ and g_j, we can expand

$$\tilde{p}' - p' = \mathrm{E}[\tilde{f}_k^2 - f_j^2] - E[\tilde{g}_k^2 - g_j^2] \tag{3.3.95}$$

$$= 2\,\mathrm{E}[(\tilde{f}_k - f_j)f_j] + 2\,\mathrm{E}[(\tilde{g}_k - g_j)g_j] + \mathrm{E}[(\tilde{f}_k - f_j)^2] + \mathrm{E}[(\tilde{g}_k - g_j)^2] \tag{3.3.96}$$

With this formula, we have

$$\mathrm{E}[(\tilde{f}_k - f_j)f_j + (\tilde{g}_k - g_j)g_j] = \frac{\tilde{p}' - p'}{2} - \frac{\mathrm{E}[(\tilde{f}_k - f_j)^2 + (\tilde{g}_k - g_j)^2]}{2} \tag{3.3.97}$$

which we insert after expanding the parentheses.

We first have

$$\mathrm{E}[(\tilde{f}_k - f_j)^2 + (\tilde{g}_k - g_j)^2] \overset{(1)}{\leq} \pi^2 \frac{\kappa^2}{t_0^2}\,\mathrm{E}[\delta_{j,k}^2(f_j^2 + g_j^2)] \overset{(2)}{<} 106\pi^2 \frac{\kappa^2}{t_0^2} p' \tag{3.3.98}$$

(1) Using Lemma 3.26 in the expanded sum for the expectation value.

(2) In the proof of (i), we have shown $\mathrm{E}[\delta_{j,k}^2] < 106$. Especially,

$$\sum_{k=0}^{T-1} |\alpha_{j,k}|^2 \delta_{j,k}^2 < 106 \tag{3.3.99}$$

for a fixed value of j. So we observe

$$\mathrm{E}[\delta_{j,k}^2(f_j^2 + g_j^2)] = \sum_{j=1}^{N} |\beta_j|^2 (f_j^2 + g_j^2) \sum_{k=0}^{T-1} |\alpha_{j,k}|^2 \delta_{j,k}^2 < 106p' \tag{3.3.100}$$

Then

$$\mathrm{E}[(\tilde{f}_k - f_j)f_j + (\tilde{g}_k - g_j)g_j] \overset{(1)}{\leq} \mathrm{E}\left[\sqrt{(\tilde{f}_k - f_j)^2 + (\tilde{g}_k - g_j)^2}\sqrt{f_j^2 + g_j^2}\right] \tag{3.3.101}$$

$$\overset{(2)}{\leq} \pi \frac{\kappa}{t_0}\,\mathrm{E}\left[|\delta_{j,k}|(f_j^2 + g_j^2)\right] \overset{(3)}{<} 11\pi \frac{\kappa}{t_0} p' \tag{3.3.102}$$

(1) Using Theorem B.6.

(2) Using Lemma 3.26.

(3) Use Theorem B.6 again, as well as $E[\delta_{j,k}^2(f_j^2 + g_j^2)] < 106p'$, for

$$\mathrm{E}[|\delta_{j,k}|(f_j^2 + g_j^2)] \leq \sqrt{E[\delta_{j,k}^2(f_j^2 + g_j^2)]E[f_j^2 + g_j^2]} < 11p' \tag{3.3.103}$$

Now for the last term involved, consider from the previous calculations

$$\tilde{p}' - p' = 2\,\mathrm{E}[(\tilde{f}_k - f_j)f_j + (\tilde{g}_k - g_j)g_j] + \mathrm{E}[(\tilde{f}_k - f_j)^2 + (\tilde{g}_k - g_j)^2] \tag{3.3.104}$$

$$< 22\pi \frac{\kappa}{t_0} p' + 106\pi^2 \frac{\kappa^2}{t_0^2} p' \leq (22\pi + 106\pi^2) \frac{\kappa}{t_0} p' \tag{3.3.105}$$

where we for now assume $t_0 \geq \kappa$. Our choice of t_0 later on will meet this condition.

We now have

$$\langle x'|\tilde{x}'\rangle > 1 - 53\pi^2 \frac{\kappa^2}{t_0^2} - 11\pi(11\pi + 53\pi^2)\frac{\kappa^2}{t_0^2} = 1 - \left(53\pi^2 + 11\pi(11\pi + 53\pi^2)\right)\frac{\kappa^2}{t_0^2} > 1 - 20000\frac{\kappa^2}{t_0^2} \tag{3.3.106}$$

yielding

$$\||x'\rangle - |\tilde{x}'\rangle\| < \sqrt{40000\frac{\kappa^2}{t_0^2}} = 200\frac{\kappa}{t_0} \tag{3.3.107}$$

(iii) Denote by $p, \tilde{p}$ the success probability of either measurement, i.e. the probabilities of measuring a 1 for the qutrit when measuring $|\varphi\rangle$ and $|\tilde{\varphi}\rangle$ respectively. We have $p \neq 0$ and especially $\tilde{p} \neq 0$ due to Theorem 3.19. Measuring after whether the qutrit becomes 1 is equivalent to measuring first wrt. to the observable differentiating between whether the qutrit assumes the state $|0\rangle$ or a state in $\mathrm{span}(\{|1\rangle, |2\rangle\})$ and then measuring for whether the qutrit becomes $|1\rangle$. Consider first, that $|x\rangle = |x'\rangle$, as well as $p = p'$ due to $g_j = 0$ for all $j \in [1, N]_{\mathbb{N}}$. Then, we further have

$$P|\tilde{x}\rangle = \sqrt{\frac{\tilde{p}'}{\tilde{p}}} P|\tilde{x}'\rangle - \sqrt{\frac{1}{\tilde{p}}} \sum_{j=1}^{N} \beta_j \sum_{k=0}^{T-1} \alpha_{j,k} \tilde{g}_k |k\rangle |v_j\rangle |2\rangle \tag{3.3.108}$$

from which we have

$$\langle x|\tilde{x}\rangle = \left\langle P|x'\rangle \middle| \sqrt{\frac{\tilde{p}'}{\tilde{p}}} P|\tilde{x}'\rangle - \sqrt{\frac{1}{\tilde{p}}} \sum_{j=1}^{N} \beta_j \sum_{k=0}^{T-1} \alpha_{j,k} \tilde{g}_k |k\rangle |v_j\rangle |2\rangle \right\rangle = \sqrt{\frac{\tilde{p}'}{\tilde{p}}} \langle P|x'\rangle | P|\tilde{x}'\rangle\rangle > 1 - 20000 \frac{\kappa^2}{t_0^2} \tag{3.3.109}$$

from which we obtain the statement directly using (ii) and $\tilde{p} \leq \tilde{p}'$.

■

Corollary 3.30 (Choosing t_0). To obtain an error smaller than ε, we set

$$t_0 \coloneqq 200 \frac{\kappa}{\varepsilon}$$

Especially, as $\varepsilon \in (0, 200)$, we have $t_0 > \kappa$.

Complexity Analysis

We analyze each step in the algorithm, starting with the initialization procedures. First, the initialization of $|b\rangle$. Denote the runtime term of $\mathcal{B}$ as $T_{\mathcal{B}}$. We make no assumptions on $T_{\mathcal{B}}$, but for a fast runtime, it should be e.g. polynomial. For the clock register initialization, we have chosen $t_0 \coloneqq 200\frac{\kappa}{\varepsilon}$, so the condition

$$t_0 < 2\pi T \tag{3.3.110}$$

for the bounds of the $\delta_{j,k}$ values as in Observation 3.10 gives $t \coloneqq \lceil \log_2(\frac{100}{\pi}\frac{\kappa}{\varepsilon}) + 1 \rceil \in \mathcal{O}(\log_2(\kappa/\varepsilon))$, or 5, which is asymptotically in the same class. The initialization of the clock register thus requires a complexity of $\mathcal{O}(t) = \mathcal{O}(\log_2(\kappa/\varepsilon))$ following Section 2.2. The QFT can be efficiently implemented and its runtime shall be omitted here. The conditional Hamiltonian simulation adds the runtime

$$\tilde{\mathcal{O}}\left(\log_2(N)\kappa s^4/\varepsilon\right) \tag{3.3.111}$$

following Section 2.7. We may note, that $\|H\| = 1$ by assumption, as we can directly calculate via the definition of the operator norm, see Theorem 2.12, that the bound on the maximal eigenvalue also bounds the operator norm this way. We omit the runtime for the qutrit rotation for generating the auxiliary qutrit in the same manner as the QFT. The last factor involved is AA, as in Algorithm 1. For that, we consider first the following theorem on the success probability.

Theorem 3.31. We have

$$0 < \tilde{p} = \sum_{j=1}^{N} \sum_{k=0}^{T-1} |\beta_j \alpha_{j,k}|^2 \tilde{f}_k^2 \in \Omega(1/\kappa^2) \tag{3.3.112}$$

Proof. As proven in Theorem 3.19, for each j, there is are approximations $k_j \in [0, T-1]_{\mathbb{N}}$ with $|\lambda_j - \tilde{\lambda}_{k_j}| < \frac{1}{4\kappa}$. So $\tilde{p} \neq 0$. Due to $\tilde{\lambda}_{k_j} \in \left(\frac{3}{4\kappa}, 1 + \frac{1}{4\kappa}\right)$ and the strict monotonicity of f in both $\left(\frac{3}{4\kappa}, 1\right]$ and $\left[1, 1 + \frac{1}{4\kappa}\right)$,

we thus have

$$\tilde{f}_{k_j}^2 > \min\left(\left\{\frac{1}{4}\sin^2\left(\frac{\pi}{2}\left(2\frac{3}{4\kappa}-1\right)\right), \frac{1}{4\kappa^2\left(1+\frac{1}{4\kappa}\right)^2}\right\}\right) = \min\left(\left\{\frac{1}{4}\cos^2\left(\frac{3\pi}{4\kappa}\right), \frac{1}{4\kappa^2\left(1+\frac{1}{4\kappa}\right)^2}\right\}\right) \tag{3.3.113}$$

$$\geq \min\left(\left\{\frac{1}{8}, \frac{1}{(25/4)\kappa^2}\right\}\right) \in \Omega(1/\kappa^2) \tag{3.3.114}$$

To bound the associated $|\alpha_{j,k}|^2$ values, we consider Theorem 3.9, which states

$$\sum_{\substack{k\in[0,T-1]_{\mathbb{N}}\\ |\delta_{j,k}|<2\pi}} |\alpha_{j,k}|^2 > 0.3 \tag{3.3.115}$$

This gives

$$\tilde{p} > \sum_{j=1}^{N} |\beta_j|^2 \sum_{\substack{k\in[0,T-1]_{\mathbb{N}}\\ |\delta_{j,k}|<2\pi}} |\alpha_{j,k}|^2 \tilde{f}_k^2 \in \Omega\left(\frac{1}{\kappa^2}\right) \tag{3.3.116}$$

which is the statement. ■

So the resulting runtime is

$$\tilde{\mathcal{O}}(\kappa(T_{\mathcal{B}} + 2\log_2(\kappa/\varepsilon) + \log_2(N)\kappa s^4/\varepsilon)) \tag{3.3.117}$$

Let us summarize the result in one theorem.

Theorem 3.32 (HHL Algorithm)**.** Let $N \coloneqq 2^n$, $n \in \mathbb{N}_{\geq 1}$. Given a well-conditioned matrix $A \in \mathbb{C}^{N\times N}$, an efficiently initializable state $|b\rangle \in \mathbb{C}^N$ with initialization complexity term $T_{\mathcal{B}}$, $|x\rangle \coloneqq \frac{1}{\|A^{-1}|b\rangle\|} A^{-1}|b\rangle$ and an error cap of $\varepsilon \in (0, 100/(4\pi))$, there is a quantum algorithm for obtaining a quantum state $|\tilde{x}\rangle \in \mathbb{C}^N$ with $\||\tilde{x}\rangle - |x\rangle\| < \varepsilon$ in time

$$\tilde{\mathcal{O}}(\kappa(T_{\mathcal{B}} + 2\log_2(\kappa/\varepsilon) + \log_2(N)\kappa s^4/\varepsilon)) \tag{3.3.118}$$

3.4 Relaxations to the Assumptions and Discussion

Relaxations We deduce possible relaxations to the multitude of assumptions presented in Section 3.1. Let $A \in \mathbb{C}^{m\times n}$ be an arbitrary matrix and $b \in \mathbb{C}^m$ be a vector, where $m, n \in \mathbb{N}_{\geq 1}$. Consider the SLE $Ax = b$, where $x \in \mathbb{C}^n$ is unknown.

1. If $b = 0$, then $x = 0$ is a solution. If $A = 0$, then if $b \neq 0$, there is no solution. Furthermore, if $\|b\| \neq 1$, then solving after $Ax = \frac{b}{\|b\|}$ gives the solution by $\|b\|x$. So we may now assume $\|b\| = 1$, $A \neq 0$ and write $|b\rangle$.

2. If $m \neq n$ and m is not a power of two, appending zero columns and rows to A and adding zero entries to $|b\rangle$ gives a sufficient form of the equation system. In the next point, we describe a reduction, which suffices for introducing Hermiticity to create a system of form $(m+n)\times(m+n)$, so the number of rows and columns for A and zero entries for $|b\rangle$ we need to add is given by $1 \leq 2^{\lceil\log_2(m+n)\rceil} - (m+n) \leq 2^{\lceil\log_2(m+n)\rceil-1} - 1$, which is polynomial. This modification further violates the invertibility requirement.

3. $|b\rangle$ must be efficiently initializable. There is no clear reduction strategy for this, except possibly preconditioning [3, p. 4], which we will also mention in connection with the sparsity.

4. We first consider the Hermiticity requirement, before the invertibility. Hermiticity is needed to approximate the eigenvalues using an element of $\frac{2\pi\mathbb{Z}}{t_0}$ each. We may skip the following reduction, if the imaginary parts of the eigenvalues are very small, and the system is already quadratic. The latter point is needed for the second previous reduction, but can also be performed by appending

additional rows or columns to the matrix and entries to the vector. In general, we can obtain a Hermitian matrix from A by performing the following reduction from [3, p. 11-12] to a system of form

$$\begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix} \begin{pmatrix} 0 \\ x \end{pmatrix} = \begin{pmatrix} |b\rangle \\ 0 \end{pmatrix} \tag{3.4.1}$$

We verify the Hermiticity by

$$\begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix}^\dagger = \begin{pmatrix} 0 & A^* \\ A^t & 0 \end{pmatrix}^t = \begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix} \tag{3.4.2}$$

The space complexity of this reduction is given by the $3mn$ additional entries to create the new matrix, but also by the n zero entries, which are added to $|b\rangle$.

We can further make statements about the eigenvalues and eigenstates of this matrix with the singular numbers and singular vectors of A. Denote the matrix appearing in the reduction SLE as $H \coloneqq |0\rangle\langle 1| \otimes A + |1\rangle\langle 0| \otimes A^\dagger$ and let the outer form of the SVD of A as in Theorem 1.13 be given by

$$A = \sum_{j=1}^{r} \sigma_j |u_j\rangle\langle v_j| \text{ and thus } A^\dagger = \sum_{j=1}^{r} \sigma_j |v_j\rangle\langle u_j| \tag{3.4.3}$$

with $r \coloneqq \mathrm{rk}(A) > 0$, as $A \neq 0$. We claim, that H has the $2r$ eigenvalues $\{\pm\sigma_j \mid j \in [1,r]_\mathbb{N}\}$ and $2r$ eigenvectors $\{|w_j^\pm\rangle \coloneqq (1/\sqrt{2})(|0\rangle|u_j\rangle \pm |1\rangle|v_j\rangle) \mid j \in [1,r]_\mathbb{N}\}$, which we can verify by direct matrix multiplication. Consider $|0\rangle|b\rangle$, as in the reduction SLE, and express it via its projection

$$\sum_{j=1}^{r} \langle |0\rangle|b\rangle ||0\rangle|u_j\rangle\rangle |0\rangle|u_j\rangle = \sum_{j=1}^{r} \langle b|u_j\rangle |0\rangle|u_j\rangle =: \sum_{j=1}^{r} \beta_j |0\rangle|u_j\rangle = \sum_{j=1}^{r} \beta_j \frac{1}{\sqrt{2}}(|w_j^+\rangle + |w_j^-\rangle) \tag{3.4.4}$$

using Lemma 3.27 and normalize it. Considerations regarding the fact, that we apply the algorithm on a projection of $|0\rangle|b\rangle$, are made in the next point. Running the HHL algorithm gives an approximation of the state

$$\sum_{j=1}^{r} \frac{\beta_j}{\sigma_j} \frac{1}{\sqrt{2}}(|w_j^+\rangle - |w_j^-\rangle) \tag{3.4.5}$$

under normalization.

5. If A is not invertible, then first need to reconsider the initial problem statement. If we have $|b\rangle \notin \mathrm{Im}(A)$, there is no target state to approximate, but if $(\lambda_1, |v_1\rangle), ..., (\lambda_N, |v_N\rangle)$ again denotes the eigenvalue-eigenstate pairs of an eigenbasis of A, then can define $P_\mathrm{Im} \coloneqq \sum_{\lambda_j \neq 0} |v_j\rangle\langle v_j|$ and $P_\ker \coloneqq \sum_{\lambda_j = 0} |v_j\rangle\langle v_j|$ to be the image and kernel projectors of A. The HHL algorithm then seemingly approximates the solution to the SLE problem

$$A|x\rangle = \frac{1}{\|P_\mathrm{Im}|b\rangle\|} P_\mathrm{Im}|b\rangle \tag{3.4.6}$$

if $P_\mathrm{Im}|b\rangle \neq 0$. But this is also not the case in general. Defining $P_w \coloneqq \sum_{\lambda_j \in [\frac{1}{\kappa}, 1]} |v_j\rangle\langle v_j|$ and $P_b \coloneqq \sum_{\lambda_j \notin [\frac{1}{\kappa}, 1]} |v_j\rangle\langle v_j|$ to be the projectors into the *well-conditioned subspace* and *bad-conditioned subspace*

$$\mathrm{span}\left(\left\{|v_j\rangle \;\middle|\; j \in [1,N]_\mathbb{N}, \lambda_j \in \left[\frac{1}{\kappa}, 1\right]\right\}\right) \text{ and } \mathrm{span}\left(\left\{|v_j\rangle \;\middle|\; j \in [1,N]_\mathbb{N}, \lambda_j \notin \left[\frac{1}{\kappa}, 1\right]\right\}\right) \tag{3.4.7}$$

and especially besides the contributions of the projection of $|b\rangle$ onto the subspace

$$\text{span}\left(\left\{|v_j\rangle \;\middle|\; j \in [1,N]_{\mathbb{N}}, \lambda_j \in \left(\frac{1}{2\kappa}, \frac{1}{\kappa}\right) \cup (1,\infty)\right\}\right) \tag{3.4.8}$$

due to the actions of the filter functions in $(1/(2\kappa), 1/\kappa)$ and afterwards, the HHL algorithm solves the SLE

$$A\,|x\rangle = \frac{1}{\|P_w\,|b\rangle\|} P_w\,|b\rangle \tag{3.4.9}$$

if $P_w\,|b\rangle \neq 0$. The filter functions still perform "true" inversion in the space spanned by the eigenstates associated with eigenvalues in $(1,\infty)$, but when these values become very large, their contributions also vanish.

Besides that, there is no division by zero inside of the algorithm, if there are eigenvalues of value zero, but the solution to the SLE may be wrong to an unacceptable degree due to the filter functions possibly filtering out most of the contribution by $P_b\,|b\rangle$. One further issue, which follows from this problem, is, that if $\langle b|v_j\rangle = 0$ for all $j \in [1,N]_{\mathbb{N}}$ with $\lambda_j \in \left(\frac{1}{2\kappa}, \infty\right)$, the algorithm does not terminate in general, as the amplitude amplification does not terminate, see Algorithm 1. The success probability of measuring a 1 must be greater than zero, as required in our analysis of the complexity.

In summary, there is no clear general strategy for generally solving this issue, and for the application of the algorithm to succeed these projections must be taken into consideration, although we also discuss strategies for scaling the eigenvalues for enlargening the well-conditioned subspace. As Harrow et al. may have implied it [3, p. 7], a person working with the algorithm may in general want to weigh the contribution of the well-conditioned and bad-conditioned subspaces first.

6. The requirement for positive-semidefiniteness mainly affects the phase estimation analysis in Section 3.3, but also the analysis of the filter functions for the overall error. As we aim to approximate $\lambda_j t_0$ for any fixed $j \in [1,N]_{\mathbb{N}}$ and $t_0 > 0$, the algorithm must allow $k < 0$. We can do that by increasing the clock register size by 1, so $t \leftarrow t+1$ in the following, and additionally applying the conditioned phase transformation

$$N\colon \mathbb{C}^T \to \mathbb{C}^T, |\tau\rangle \mapsto e^{i\frac{\tau}{T}\cdot 2\pi\left(\frac{T}{2}-1\right)}\,|\tau\rangle \tag{3.4.10}$$

on the clock register before the application of $\text{QFT}^\dagger$ in Algorithm 2. This map is unitary and efficiently implementable, as considering the binary representation $\tau = \tau_{t-1}...\tau_0 = \sum_{i=0}^{t-1}\tau_i 2^i$ of an index $\tau \in [0, T-1]_{\mathbb{N}}$ gives

$$N = \bigotimes_{i=0}^{t-1} \begin{pmatrix} e^{\frac{0}{T}\cdot 2\pi\left(\frac{T}{2}-1\right)} & 0 \\ 0 & e^{\frac{2^{(t-1)-i}}{T}\cdot 2\pi\left(\frac{T}{2}-1\right)} \end{pmatrix} \tag{3.4.11}$$

The state after the phase estimation using this phase transformation is thus

$$\frac{\sqrt{2}}{T}\sum_{j=1}^{N}\beta_j \sum_{k=0}^{T-1}\left(\sum_{\tau=0}^{T-1}\sin\left(\frac{\pi(\tau+\frac{1}{2})}{T}\right)e^{\frac{i\tau}{T}\left(\lambda_j t_0 - 2\pi\left(k-\left(\frac{T}{2}-1\right)\right)\right)}\right)|k\rangle\,|v_j\rangle\,|0\rangle \tag{3.4.12}$$

This gives

$$k - \left(\frac{T}{2}-1\right) \in \left[-\left(\frac{T}{2}-1\right), \frac{T}{2}\right] \tag{3.4.13}$$

Modifying the algorithm in this way does not change any of the results from the phase estimation analysis, but it allows the existence of a k with $|\delta_{j,k}| < 2\pi$ for any j. These approximations should

then be denoted by $\tilde{\lambda_k} \coloneqq \frac{2\pi}{t_0}(k-(T/2-1))$. We uncompute the registers analogously. We further need to mirror the filter functions at $y=0$ to allow for negative approximations, similarly to Childs et al. [40, p. 5]. The statements of the Lipschitz-continuity of $|h(\cdot)\rangle$ in Section 3 and the proof of the special Lipschitz-type condition in Lemma 3.26 do not change either, because for the first we only considered the derivatives and for the second, we mirrored the filter functions. And, since Theorem 3.29 and the complexity analysis are also not affected, this reduction solves the problem.

7. Here we have no clear reduction of the sparsity of A. Possible techniques may include performing basis switches or simply considering simulation techniques for different Hamiltonians[7] following [3, p. 6] or even allowing the use of a *preconditioner* [3, p. 4], which would make the problem more suitable for the HHL algorithm.

8. Efficient row-computability is essential to the Hamiltonian simulation time. We refer to the last point.

9. For $\frac{1}{\kappa} \leq \lambda_j \leq 1$ for all $j \in [1,N]_{\mathbb{N}}$, we first consider in the positive-semidefinite case, that the contribution of the space spanned by the eigenstates with eigenvalue $\lambda_j > 1$ vanishes with larger λ_j in the inversion by the filter function f. If we still need this requirement, for instance since many eigenvalue contributions come from eigenvalues in $(0, 1/(2\kappa)]$, then we may consider calculating the maximal condition number $\lambda_{\max} \in \mathbb{R}_{>0}$ and then by $A\frac{1}{\lambda_{\max}}x = |b\rangle$, we have indeed this requirement for the eigenvalues in this SLE. This is due to our ability to scale the Hamiltonian, which in return scales the eigenvalues due to $(\mu A)\,|v_j\rangle = (\mu\lambda_j)\,|v_j\rangle$ for any $\mu \in \mathbb{R}$, $j \in [1,N]_{\mathbb{N}}$. We recover the solution vector for the original system by multiplying with $\lambda_{\max}$. When using an upper bound, we must consider, if our approximation or upper bound for κ does not cut off eigenvalues, which become very small by dividing by $\lambda_{\max}$. For $\lambda_{\min} = 1$ or $\lambda_{\min} \approx 1$, it would also suffice to use κ as this maximum eigenvalue bound. If we have only negative eigenvalues, we can multiply A with -1 and in case of having both positive and negative eigenvalues, we cannot additively shift the eigenvalues, so the only clear reduction strategy would be to weigh both subspaces and multiply by -1, iff the subspace spanned by the eigenstates of the negative eigenvalues has a larger contribution for the accuracy of the solution.

10. We may omit the requirement for having the exact value of κ by providing an upper bound on it, similarly to Childs et al. [40, p. 3]. This affects mainly the filter functions and the algorithm runtime. How such an upper bound can be obtained is generally unclear, but we have seen one instance, where one can do that, when we discussed the example of the condition number of *Hilbert matrices*, see Example 1.26. Another strategy involves increasing κ exponentially by $1, 2, 4, ...$, as [3, p. 6] suggest. For that, we would still need to have a success probability of the measurement, which is not equal to zero, as otherwise AA does not terminate.

Discussion and Outlook Our description of the original HHL algorithm is finished. We want to now discuss some chosen aspects.

We first want to give some remarks regarding the original HHL paper and why we needed some more technical lemmas in this thesis. First, the initialization of the clock state is not further elaborated, only a small mention of it is made in [3, p. 2]. As a sidenote, we may consider it to be interesting, that we can obtain the formula for the antiderivative using the formula by Brassard et al.. The calculation of the alternative representation of the $\alpha_{j,k}$ values in Lemma 3.12 contains mistakes, due to which the result is off by $-1/2$. The phase estimation analysis presented [3, pp. 10-11] concludes with a bound of $|\alpha_{j,k}| \leq \frac{8\pi}{\delta_{j,k}^2}$, using the lower bound in Lemma 3.11, but it is not applicable, as, see our argument immediately after Observation 3.10, the assumption $|\delta_{j,k}| \leq T/10$ is generally wrong and thus the arguments of the sine functions can be in $[3.5\pi/4, \pi]$, which supersedes the root $\sqrt{6}$ of the lower bounding polynomial $x - x^3/6$, which leads to both terms of the product $\sin\left(\frac{\delta_{j,k}+\pi}{2T}\right)\sin\left(\frac{\delta_{j,k}-\pi}{2T}\right)$ in the denominator becoming negative. Furthermore, in [3, p. 6], a description of the actual implementation of qutrit rotation is missing.

[7] In this sense, we can consider the Hamiltonian simulation as replaceable.

The choice of initial coefficients for the clock register is seemingly arbitrary. Harrow et al. reason, that it suites the error analysis well [3, p. 2], which it does, but the optimality has not been shown. From our research, it appears, that this aspect of the algorithm has not been further studied. Some authors instead suggest in their diagrams, that we can initialize the register by applying the Hadamard transformation. Examples of this constitute [12, p. 30], [41, p. 351] and [42, p. 5]. It would seem, that most experimental setups prefer this initialization procedure, although it does not conform to the original description and especially does not fulfill the requirements for the guaranteed bounds from the error analysis.

Some experimental implementations of the algorithm further fix the evolution time to 2π, see [43, p. 4-5] for a demonstration. This parameter can, in most cases, be chosen in a better way, as we argue in the next example. In the example of Cao et al., they also do not initialize using the clock register using the proposed coefficients, and they do not use the filter functions. Although in this case, this could be due to the example being of pure demonstrational nature. In general, the choices of t and t_0 do not need to be fixed, especially, if we possibly consider a nyche case.

Example 3.33. Suppose all eigenvalues are in $\mathbb{N}_{\geq 1}$. Then we can choose t at least large enough, s.t. $T - 1 \geq \max\{\lambda_1, ..., \lambda_N\}$ and $t_0 := 1$. In this case, the eigenvalue estimation is near perfect for all eigenvalues. Suppose $\lambda_j = k$, then $\delta_{j,k} = 0$ and, by Lemma 3.12, we have

$$|\alpha_{j,k}| = \frac{\sqrt{2}}{T} \frac{1}{\sin\left(\frac{\pi}{2T}\right)} > \frac{\sqrt{2}}{T} \frac{2T}{\pi} = \frac{2\sqrt{2}}{\pi} > 0.9003 \tag{3.4.14}$$

under the use of Lemma 3.11.

To improve numerical stability, specific filter functions were chosen. The authors state, that the choice of f and g is arbitrary [3, p. 6]. This aspect could be studied further to find out, if the numerical precision can be improved by choosing different functions. We also argue, that a straight cut-off at the boundary $\frac{1}{\kappa}$ would suffice for most needs. A dive into the referred literature at [3, p. 6] may also help improve the general understanding regarding this point.

We want to again consider the initial problem statement for the HHL algorithm. The goal was to solve an SLE. After we have seen the actual operations of the HHL algorithm, it is valid to ask what exact answer the HHL algorithm is approximating. In our discussion of possible relaxation techniques for the invertibility of A, see 5., we observed, that the result of the algorithm is an inversion of the projection of the given vector into the so-called well-conditioned subspace of eigenstates of A. The result of the algorithm is thus not always the actual solution to the SLE itself. We know, that in general, there may be no solutions, exactly one or several more, neither case can always be determined using the HHL algorithm. In some cases, as in Chen and Gaos algorithm for obtaining solutions of polynomial equation systems, which we present in Section 4.4, one can decide the solvability by studying a special case and more techniques.

Furthermore, the result of the HHL algorithm is a quantum state, not a classical vector. The no-cloning-theorem [19, pp. 81-84], in practice, thus makes the duplication of the result state impossible. Obtaining the entries of a quantum state is in general not a trivial task, although there have been suggestions for such methods [28, pp. 14-25]. HHL can thus only be used as a subroutine in a larger quantum or hybrid quantum algorithm, which makes its applicability in some fields, such as machine learning or cryptography, difficult.

We believe, that the given runtime for the Hamiltonian simulation in [3, p. 6] is off by a factor of $\mathcal{O}(s^2)$. We have detailled our calculation for the runtime of the method used in Section 2.7, while in the original HHL paper, there is no mention of where to find or derive the claimed runtime. The researchers Childs et al., who are actively involved in the research of Hamiltonian simulation, as for instance [40, 44] show, have further supported this claim in [44, p. 2]. If this finding is true, then the runtime of the HHL algorithm is as analyzed approximately $\tilde{\mathcal{O}}(\log_2(N)\kappa^2 s^4/\varepsilon)$ and not $\tilde{\mathcal{O}}(\log_2(N)\kappa^2 s^2/\varepsilon)$, which could have far reaching consequences regarding the research applying the HHL algorithm, of which we mention some recent results in the next paragraph.

In the next subsection, we will discuss two improvements of the algorithm by Ambainis and Childs et al.. To add to this, we have in general found, that since the release of the paper, the algorithm has been extended and applied by multiple authors, as Harrow et al. themselves have observed [3, pp. 4-5]. For instance by using it to solve *non-linear differential equations* [45], as such problems can often be reduced to linear systems [8]. [46] presents four more improvements in its introduction and the *Variational Quantum Linear Solver* (VQLS), an algorithm, that combines a minimization problem for a cost function on a classical computer with a quantum algorithm for solving SLEs. Hybrid quantum-classical algorithms have been developed and tested, one such test constitutes [42], thus giving a different class of quantum SLE solvers than HHL-inspired ones.

3.5 Outline of Two Improvements

We want to briefly mention two improved algorithms based on the ideas of HHL.

Variable Time Amplitude Amplification

Ambainis suggested a model of gate quantum computation, in which a quantum algorithm could "halt" at different times, and derived another algorithm for solving linear systems of equations based on HHL [47]. We will explain the model briefly and then summarize the result.

Branched Quantum Computations The model of VTAA [47, pp. 5-8] is based on a state space of form $\mathcal{H} \coloneqq \mathbb{C}^3 \otimes \mathcal{H}_o$, where $\mathcal{H}_o$ is a Hilbert space. The system, or our register, shall be in the states $|\psi_1\rangle, ..., |\psi_m\rangle \in \mathcal{H}$ at times $t_1, ..., t_m \in \mathbb{R}_{\geq 0}$. Every t_i, $i \in [1, m]_{\mathbb{N}}$, the system may *stop*, meaning that we stop the algorithm with a given probability $p_i \in [0, 1]$. We require

1. that there are subspaces $\mathcal{H}_i$ of $\mathcal{H}_o$ forming an ascending chain $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq ... \subseteq \mathcal{H}_m = \mathcal{H}_o$.
2. that we can express $|\psi_i\rangle$ for an $i \in [1, m]_{\mathbb{N}}$ as
$$|\psi_i\rangle = \alpha_{i,0} |0\rangle \otimes |\psi_{i,0}\rangle + \alpha_{i,1} |1\rangle \otimes |\psi_{i,1}\rangle + \alpha_{i,2} |2\rangle \otimes |\psi_{i,2}\rangle \tag{3.5.1}$$
where $\alpha_{i,0}, \alpha_{i,1}, \alpha_{i,2} \in \mathbb{C}$ are scalars and $|\psi_{i,0}\rangle, |\psi_{i,1}\rangle \in \mathcal{H}_i$, as well as $|\psi_{i,2}\rangle \in \mathcal{H}_o \cap \mathcal{H}_i^{\perp}$ may not necessarily be valid quantum states. We further have $|\psi_{m,2}\rangle = 0$.
3. Let $i \in [1, m-1]_{\mathbb{N}}$ be fixed. Denote by $P_{\mathcal{H}_i}$ the projector into $\mathcal{H}_i$. Then
$$|\psi - i, 0\rangle = P_{\mathcal{H}_i} |\psi_{i+1,0}\rangle \wedge |\psi - i, 1\rangle = P_{\mathcal{H}_i} |\psi_{i+1,1}\rangle \tag{3.5.2}$$

We may interpret this model the following way: During computation, parts of our state switch the Hilbert space, in which the main calculation is currently performed. By doing that, we are enabling the possibility of dovetailing, i.e. interweaving multiple computations. The interpretation of the states comes from requirement 2: Similar to the HHL algorithm, $|\psi_{i,1}\rangle$ decodes the "good" part of the state and $|\psi_{i,0}\rangle, |\psi_{i,2}\rangle$ decode the bad parts of the state. The desired results of the computation itself are thus contained inside of $\mathcal{H}_i$ at time t_i respectively.
The times $t_1, ..., t_m$ and probabilities $p_1, ..., p_m$ were let lose by us and are up to the algorithm designer to choose. As Ambainis, one may define the *average stopping time* by

$$T_a \coloneqq \sqrt{\sum_{i=1}^{m} p_i t_i^2} \tag{3.5.3}$$

and set the maximum time $T_M \coloneqq t_m$ and the success probability at time point t_m to be $p_s \coloneqq |\alpha_{m,1}|^2$. Ambainis then proves the following theorem.

Theorem 3.34. There is a quantum algorithm, which amplifies the success probability of an algorithm in the VTAA model to give a successful measurement in time

$$\mathcal{O}\left(T_M \log^{0.5}(T_M) + \frac{T_a}{p_s} \log^{1.5}(T_M)\right) \tag{3.5.4}$$

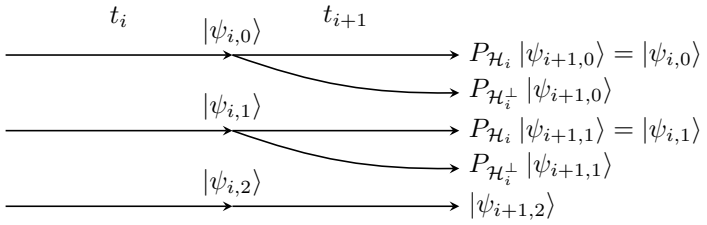


Figure 14: Illustation of the difference between the three components of two different consecutive states in a VTAA algorithm, where $i \in [1, m]_{\mathbb{N}}$ is fixed. The branches in the arrows indicate sums, i.e. e.g. $|\psi_{i,0}\rangle = P_{\mathcal{H}_i} |\psi_{i,0}\rangle + P_{\mathcal{H}_i^\perp} |\psi_{i,0}\rangle$.

The main result by Ambainis is now, that by expressing the HHL algorithm inside of the VTAA model, we can improve the success probability from $\mathcal{O}(\kappa^2)$ wrt. the runtime factor of κ. There we had $\mathcal{O}(\kappa^2)$ as the dependence, which was obtained by choosing the evolution time in dependence of $\mathcal{O}(\kappa)$ and then running AA for another dependence on $\mathcal{O}(\kappa)$. Ambainis acquires the following result [47, pp. 8-12].

Theorem 3.35. Using VTAA, there is a quantum algorithm, which improves the runtime of the HHL algorithm to

$$\tilde{\mathcal{O}}\left(\log_2(N)\kappa \log_2^3\left(\frac{\kappa}{\varepsilon}\right) s^2 \log_2^2\left(\frac{1}{\varepsilon}\right)\frac{1}{\varepsilon^3}\right) \tag{3.5.5}$$

The runtime cited comes from the fact, that the phase estimation procedure of runtime $\mathcal{O}(\log_2(N)\kappa s^2/\varepsilon)$ is used as a subprocedure of the algorithm [47, p. 9], but Ambainis omits the $\mathcal{O}(\log_2(N)s^2)$ factor in [47, p. 12]. Note, that we cited the runtime factor $\mathcal{O}(s^2)$ to conform with these papers, although it should be $\mathcal{O}(s^4)$.

Remark 3.36. We may note, that while the dependence on κ is better than the original HHL algorithm, the error dependence is significantly worse.

Fourier Decompositions for Sublinear Error Dependence

In a 2015 paper, Childs et al. presented three approaches [40] to substantially improving the error dependence of the HHL algorithm. We shortly describe the so-called *Fourier approach*, which we shall divide into the explanation of three conceptual steps.

- First, the results include the use of newer techniques for the Hamiltonian simulation, as presented by Childs et al. in [30].
- Secondly, one important aspect of the paper is the use of LCUs as in [40, pp. 5-8]. Let $A \in \mathbb{C}^{N\times N}, N \coloneqq 2^n, n \in \mathbb{N}_{\geq 1}$ be the matrix of the SLE. Assume further it is, possibly after a reduction, Hermitian and assume for the sake of the conceptual overview, that it is invertible. The idea is then to, similiarly to the Hamiltonian decomposition in Section 2.7, decompose the matrix A^{-1} into a unitary sum and to simulate the sum. The unitaries chosen are indeed e^{iAt_j}, where $t_j \in \mathbb{R}$ are times, giving a decomposition of form $A^{-1} = \sum_j \alpha_j e^{iAt_j}$ with coefficients $\alpha_j \in \mathbb{C}$. By performing a basis switch, the authors then reduce the problem of approximating this decomposition to approximating a real univariate decomposition $x^{-1} = \sum_j \alpha_j e^{ixt_j}$ for $x \in [-1, -1/\kappa] \cup [1/\kappa, 1]$.
- Thirdly, to compute the aforementioned decomposition of $1/x$, the following Fourier transformation is used:

$$\frac{1}{x} = \frac{i}{\sqrt{2\pi}} \int_0^\infty \int_{-\infty}^\infty z e^{-z^2/2} e^{-ixyz}\, dz\, dy \tag{3.5.6}$$

  As in [40, pp. 10-11]. It is shown how to discretize these integrals, giving a suitable quantum algorithm for approximating A^{-1}.

We recognize again the pattern of using an efficient decomposition of the initial linear operator for solving the SLE problem. Childs et al. then have as one of their results the following theorem as in [40, p. 4].

Theorem 3.37 (Fourier Approach to HHL)**.** Using more recent results for Hamiltonian simulation, techniques involving LCUs and Fourier transformations, there is a quantum algorithm for solving an SLE in time

$$\mathcal{O}\left(s\kappa^2 \log_2^{2.5}\left(\frac{\kappa}{\varepsilon}\right)\left(\log_2(N) + \log_2^{2.5}\left(\frac{\kappa}{\varepsilon}\right)\right)\right) \tag{3.5.7}$$

The other two approaches include the use of *Chebyshev polynomials* and the modification of Ambainis' VTAA HHL algorithm. The three approaches are not equivalent, as pointed out in [40, p. 4] and have their own advantages and disadvantages, which we shall not elaborate, as this is a high level overview of the results.

4 Application on the Cryptanalysis of AES

The AES, synonymously *Rijndael*, is a famous, widely used block cipher. It is specified by the US-American NIST in [48]. AES is a symmetric cipher, meaning that it uses one key for the encryption and decryption of blocks. The key is K bits long, where $K \in \{128, 192, 256\}$. Thus, it is clear that for a brute force approach to key retrieval with Grovers algorithm, one can achieve a quadratic improvement from a runtime of $\mathcal{O}(2^K)$ to $\mathcal{O}(2^{K/2})$.
Rijndael and AES are two different cipher specifications. As described by the original authors of both ciphers [49, p. 31], the difference lies in the allowed values of the input block length and the cipher key length. We will focus on AES, as it is the cipher of our cryptanalytical interest.
The goal of this last subsection is twofold. For one, we want to discuss the inner workings of Rijndael and its formulation as a so-called BES-cipher. Especially, we want to form a system of equations for key recovery using that, which we will however not solve, as this is not the scope of this thesis. Secondly, we will discuss current research on this topic in the context of two recent papers by the researchers Chen and Gao [4] and Ding et al. [5]. We chose BES, as it overcomes a small algebraic problem when attempting to formulate such an equation system with a comparatively simple solution. We furthermore analyze the size of the system.
AES has proven to be a reliable cipher over the years, resisting any attempt at successful cryptanalysis yet, as a survey by Nover shows [50]. The authors of Rijndael, Joan Daemen and Vincent Rijmen, released a book on the details and the design philosophy of Rijndael [49], as referenced above. It shall be our main source for the next subsection, next to the FIPS cipher specification.

4.1 An Algebraic Description of AES

An overview of AES is given in Figure 15. Table 1 lists the relevant parameters. We shall use the symbol K for the key itself. One difficulty in this description is differentiating between the different representations of bytes: A byte can be seen as a vector from the vector space $\mathrm{GF}(2)^8$, an integer from the finite modulo ring $\mathbb{F}_{2^8}$ or as a polynomial from the field $\mathrm{GF}(2)[x]/(p)$, where $p \in \mathrm{GF}(2)[x]$ is an irreducible polynomial, see Corollary 1.31. We shall explicitely state the form we use each time. We always index starting from the least significant bit, so for instance we may have $b = b_7b_6b_5b_4b_3b_2b_1b_0 = 01110010$, which corresponds to $(0, 1, 0, 0, 1, 1, 1, 0)^t$, 114 or $x^6 + x^5 + x^4 + x$.
The input of the algorithm is both a $32N_b$-bit plaintext $P \in \mathbb{F}_{2^8}^{4\times N_b}$ and a $32N_k$-bit key $K \in \mathbb{F}_{2^8}^{4\times N_k}$. The output is a $32N_b$-bit long encrypted block $C \in \mathbb{F}_{2^8}^{4\times N_b}$. We will describe each step of the AES in detail and algebraically. We also do not fix N_b, N_r or N_k, as it is not necessary for our discussion.
One major design criterion of AES was space-efficiency [49, pp. 4-5], thus, we do not require a lot of storage. We work with a null-indexed column-major enumeration of the input plaintext bytes along a $4 \times N_b$ grid following [48, p. 9]. Let the plaintext be the initial *state* $S := (s_{(i-1)(j-1)})_{i,j\in 4\times N_b} \in \mathbb{F}_{2^8}^{4\times N_b}$ of the current encryption or decryption. The plaintext indices start at the first byte, independent of endianness. In other words, $S = P$ and, since $P = (p_0, ..., p_{4N_b-1}) \in \mathbb{F}_{2^8}^{4N_b} \cong \mathbb{F}_{2^8}^{4\times N_b}$, we can write for both P and K analogously

$$P = S = \begin{pmatrix} p_0 & p_4 & p_8 & p_{12} \\ p_1 & p_5 & p_9 & p_{13} \\ p_2 & p_6 & p_{10} & p_{14} \\ p_3 & p_7 & p_{11} & p_{15} \end{pmatrix} \qquad K = \begin{pmatrix} k_0 & k_4 & k_8 & k_{12} \\ k_1 & k_5 & k_9 & k_{13} \\ k_2 & k_6 & k_{10} & k_{14} \\ k_3 & k_7 & k_{11} & k_{15} \end{pmatrix} \tag{4.1.1}$$

for the initial state [49, p. 33], here for the case $N_b = N_k = 4$.

Parameter	Meaning	AES-128, AES-192, AES-256
N_b	Block length in 32-bit words.	4, 4, 4
K	Length of cipher key in bits.	128, 192, 256
N_k	Key length in 32-bit words.	4, 6, 8
N_r	Round count.	10, 12, 14

Table 1: AES Parameters, according to [48, pp. 13-14].

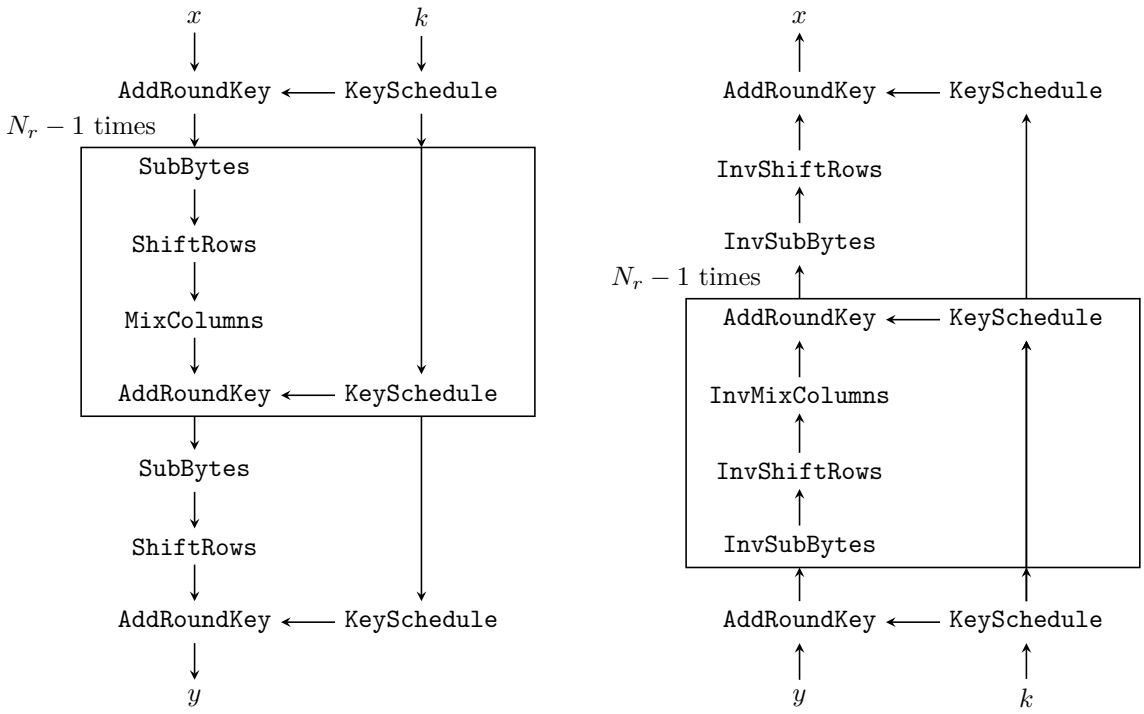


Figure 15: AES encryption and decryption block diagram. The inverse versions of the encryption functions are defined in analogy to them, and will not be of concern to us.

(i) `SubBytes` [49, pp. 34-37]: Each byte in the state is interpeted as an element of the field $F \coloneqq \mathrm{GF}(2)[x]/(x^8 + x^4 + x^3 + x + 1)$. First, consider the so-called *patched inverse* bijection ι, as well as the matrix $L_A \in \mathrm{GF}(2)^{8\times 8}$ and vector $v_A \in \mathrm{GF}(2)^8$:

$$\iota\colon F \xrightarrow{\cong} F, a \mapsto \begin{cases} 0 & a = 0 \\ a^{-1} & a \neq 0 \end{cases} \qquad L_A \coloneqq \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \qquad v_A \coloneqq \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \tag{4.1.2}$$

The `SubBytes` step performs the map

$$\rho\colon F \to F, a \mapsto L_A \iota(a) + v_A \tag{4.1.3}$$

for each byte s in S. This part of the description already poses a problem for the cryptanalysis of the cipher, as we switched from F to $\mathrm{GF}(2)^8$ for the application of the affine transformation. ρ is also called the *Rijndael S-Box*. We may also note, that the byte is interpreted as a column vector with the top entry being the LSB. L_A is further invertible, as $\det(L_A) = 5$, which we may check in a long calculation via the recursive development of the determinant.

(ii) `ShiftRows` [49, pp. 37-38]: Use the following map:

$$\begin{pmatrix} s_{00} & s_{01} & s_{02} & s_{03} \\ s_{10} & s_{11} & s_{12} & s_{13} \\ s_{20} & s_{21} & s_{22} & s_{23} \\ s_{30} & s_{31} & s_{32} & s_{33} \end{pmatrix} \mapsto \begin{pmatrix} s_{00} & s_{01} & s_{02} & s_{03} \\ s_{11} & s_{12} & s_{13} & s_{10} \\ s_{22} & s_{23} & s_{20} & s_{21} \\ s_{33} & s_{30} & s_{31} & s_{32} \end{pmatrix} \tag{4.1.4}$$

We can also express this operation via a permutation matrix $M_A \in F^{4N_b \times 4N_b}$. The above instruction is then equivalent to taking a row-major enumeration of S and performing

$$S \mapsto S_A S \tag{4.1.5}$$

where we denote with the notation of a permutation, i.e. each entry shows the index of the 1-entry in the column

$$S_A \coloneqq (0, 5, 10, 15, 4, 9, 14, 3, 8, 13, 2, 7, 12, 1, 6, 11) \tag{4.1.6}$$

(iii) `MixColumns` [49, pp. 39-41]: The $i \in [0, N_b - 1]_{\mathbb{N}}$th column vector $(s_{ij})_{j \in [0,3]_{\mathbb{N}}}$ of S is treated as a vector from $F' \coloneqq \mathbb{F}_{2^8}[x]/(x^4+1)$ and multiplied with $3x^3 + x^2 + x + 2$. This is equivalent to taking a column-major enumeration of S and applying the diagonal matrix $M_A \coloneqq \mathrm{diag}(C_A, C_A, C_A, C_A)$, where

$$C_A \coloneqq \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix} \tag{4.1.7}$$

following Example 1.32. Note, that the first byte in a column thus corresponds to the coefficient x^3 in each polynomial, and so on.

(iv) `AddRoundKey` [49, p. 41]: During the $i \in [0, N_r]_{\mathbb{N}}$th round, add the current round key, so $S \mapsto S + \tilde{K}_i$. See (v).

(v) `KeySchedule` [49, pp. 43-46]: The algorithm initially creates $N_r + 1$ additional keys $\tilde{K}_0, ..., \tilde{K}_{N_r} \in \mathbb{F}_{2^8}^{4 \times N_b}$ for the `AddRoundKey` step. This procedure is called `KeyExpansion`. Let $W \coloneqq \mathbb{F}_{2^8}^{4 \times N_b (N_r+1)}$ be a matrix of $N_b(N_r+1)$ 32-bit words. Four columns each correspond to a round key. We further define so-called *round constants* $r_{i+1} \coloneqq x^i \in F$, $i \in \mathbb{N}$. All additions are performed in $\mathrm{GF}(2)^8$, so by a component-wise exclusive-or operation.

There are versions of the `KeyExpansion` for $N_k \leq 6$ and $N_k > 6$. For AES, these versions correspond to the cases $N_k \in \{4, 6\}$ and $N_k = 8$. If $N_k \leq 6$, then the matrix W is constructed column by column according to the following rules, in this order of precedence:

$$\begin{array}{lll} w_{ij} & \coloneqq k_{ij} & i \in [0,3]_{\mathbb{N}}, j \in [0, N_k - 1]_{\mathbb{N}} \\ w_{0j} & \coloneqq w_{0(j-N_k)} + \rho(w_{1(j-1)}) + r_{j/N_k} & j \in [N_k, N_b(N_r+1) - 1]_{\mathbb{N}}, j = 0 \bmod N_k \\ w_{ij} & \coloneqq w_{i(j-N_k)} + \rho(w_{((i+1) \bmod 4)(j-1)}) & i \in [1,3]_{\mathbb{N}}, j \in [N_k, N_b(N_r+1) - 1]_{\mathbb{N}}, j = 0 \bmod N_k \\ w_{ij} & \coloneqq w_{i(j-N_k)} + w_{i(j-1)} & i \in [0,3]_{\mathbb{N}}, j \in [N_k, N_b(N_r+1) - 1]_{\mathbb{N}}, j \neq 0 \bmod N_k \end{array} \tag{4.1.8}$$

If $N_k > 6$, then the matrix W is constructed in a similar way, that is according to:

$$\begin{array}{lll} w_{ij} & \coloneqq k_{ij} & i \in [0,3]_{\mathbb{N}}, j \in [0, N_k - 1]_{\mathbb{N}} \\ w_{0j} & \coloneqq w_{0(j-N_k)} + \rho(w_{1(j-1)}) + r_{j/N_k} & j \in [N_k, N_b(N_r+1) - 1]_{\mathbb{N}}, j = 0 \bmod N_k \\ w_{ij} & \coloneqq w_{i(j-N_k)} + \rho(w_{((i+1) \bmod 4)(j-1)}) & i \in [1,3]_{\mathbb{N}}, j \in [N_k, N_b(N_r+1) - 1]_{\mathbb{N}}, j = 0 \bmod N_k \\ w_{ij} & \coloneqq w_{i(j-N_k)} + \rho(w_{i(j-1)}) & i \in [0,3]_{\mathbb{N}}, j \in [N_k, N_b(N_r+1) - 1]_{\mathbb{N}}, j = 4 \bmod N_k \\ w_{ij} & \coloneqq w_{i(j-N_k)} + w_{i(j-1)} & i \in [0,3]_{\mathbb{N}}, j \in [N_k, N_b(N_r+1) - 1]_{\mathbb{N}}, j \neq 0 \bmod N_k \end{array} \tag{4.1.9}$$

After W is obtained, we derive the keys by $W = \begin{pmatrix} \tilde{K}_0 & \tilde{K}_1 & ... & \tilde{K}_{N_r} \end{pmatrix}$.

The output of the algorithm is stored in S. We may also note, that all steps described are invertible, allowing decryption.

4.2 The BES Cipher

The cryptanalysis of AES in form of the above algebraic description is complicated, as we have to switch between the field F and the vector space $\mathrm{GF}(2)^8$. The *Big Encryption System* (BES) cipher by Murphy and Robshaw [51] defines a family of ciphers similar structure, and simplifies the cryptanalysis by only using operations in F. We shall describe the BES for every version of AES and without the so-called modified key schedule [51, p. 3].[8]

We define the map

$$\phi\colon F \to F^8 \tag{4.2.1}$$

$$b \mapsto \left(b^{(2^0)}, b^{(2^1)}, ..., b^{(2^7)}\right) \tag{4.2.2}$$

For any $n \in \mathbb{N}_{\geq 1}$, ϕ is extended component-wise, giving

$$\phi_n\colon F^n \to F^{8n} \tag{4.2.3}$$

$$b \mapsto (\phi(b_1), \phi(b_2), ..., \phi(b_n)) \tag{4.2.4}$$

Analogously, we define the matrix operation $\phi_{m\times n}$, $m \in \mathbb{N}_{\geq 1}$, via ϕ_{mn}. We further let

$$\iota(\phi(a)) \coloneqq \phi(\iota(a)) \tag{4.2.5}$$

with ι being the patched inverse, as described in the AES `SubBytes` operation, see (i) in the previous subsection. The component-wise application of ι gives the analogous definitions for the general functions ϕ_n and $\phi_{m\times n}$.

Theorem 4.1 (Properties of ϕ)**.** ϕ is injective and additive. Both properties carry over to ϕ_n. [9]

Proof. The injectivity can be read off by observing the behavior on the first component. For the additivity we calculate in $\mathrm{GF}(2)[X]$ and $k \in [0,7]_{\mathbb{N}}$ using the binomial development:

$$(a+b)^{(2^k)} = \sum_{l=0}^{2^k} \binom{2^k}{l} a^l b^{2^k-l} = a^{(2^k)} + b^{(2^k)} \tag{4.2.6}$$

Note that the terms for $1 \leq l \leq 2^k - 1$ vanish as the binomial factors are natural numbers and especially divisible by 2. We slightly abuse the notation here, but the point is, that the terms in the middle of the sum are evenly often added together, which leads to the polynomial powers vanishing. ■

As mentioned, the main result by Murphy and Robshaw is, that we can represent the AES algorithm by only using operations in F. It is not yet clear, why ϕ could help our cause. In the same manner as in Section 4.1, we shall describe all five operations in the language of BES. For that, we first map the input plaintext-key pair (P, K) via ϕ into the BES-associated vector spaces. So now, denote $P \in \left(F^8\right)^{4\times N_b}$, $K \in \left(F^8\right)^{4\times N_k}$ and for the state $S \in \left(F^8\right)^{4\times N_b}$. Also $C \in \left(F^8\right)^{4\times N_b}$, as we will see.

(i') `SubBytes` [51, pp. 5-8]: The patched inverse ι can be applied component-wise for each entry s_{ij} in S. In the original AES, we applied the operator $L_A\colon \mathrm{GF}(2)^8 \to \mathrm{GF}(2)^8$ after ι and then added v_A. This is an operation in $\mathrm{GF}(2)^8$, not F, and it is not clear, how we could represent this operation with a linear transformation. Define

$$\psi\colon F \to \mathrm{GF}(2)^8, \sum_{k=0}^{7} b_k x^k \mapsto \begin{pmatrix} b_0 \\ ... \\ b_7 \end{pmatrix} \tag{4.2.7}$$

[8] As a sidenote, that the authors of AES proposed a similar method as described here under the name of *AES-GF*, see [49, pp. 192-194].

[9] Field theorists may be reminded of the *Frobenius homomorphism* [17, p. 337]. The additivity there, the *Frobenius rule*, is also called *the freshmans dream*. The proof idea for the additivity is the same.

to be the bijective natural embedding of F into $\mathrm{GF}(2)^8$. Then, we form the map $f \coloneqq \psi^{-1} \circ L_A \circ \psi$. For the following derivation, we want a polynomial, which interpolates f. The Lagrangian interpolation method [17, p. 193] gives

$$f(x) = \sum_{b \in F} f(b) \prod_{c \in F \setminus \{b\}} \frac{x-c}{b-c} = \sum_{k=0}^{7} \lambda_k x^{(2^k)} \tag{4.2.8}$$

with $(\lambda_i)_{i \in [0,7]_\mathbb{N}} \coloneqq (\texttt{05}, \texttt{09}, \texttt{f9}, \texttt{25}, \texttt{f4}, \texttt{01}, \texttt{b5}, \texttt{8f})$ in hexadecimal notation following [51, p. 7], where we do not verify this result here, as this would require a large computation using a computer, for which we know the needed multiplication techniques. L_A is invertible and thus f. Interpret the hexadecimal notation here digit-wise, so e.g. $\texttt{f9} = 11111001$.

Similarly to ϕ, f is additive. For the matrix representing the action of the linear map L_A, we may choose

$$\hat{L}_B \coloneqq \left(\lambda_{j-1}^{(2^{i-1})}\right)_{i,j \in 8 \times 8} = \begin{pmatrix} \lambda_0^{(2^0)} & \lambda_1^{(2^0)} & \lambda_2^{(2^0)} & \lambda_3^{(2^0)} & \lambda_4^{(2^0)} & \lambda_5^{(2^0)} & \lambda_6^{(2^0)} & \lambda_7^{(2^0)} \\ \lambda_7^{(2^1)} & \lambda_0^{(2^1)} & \lambda_1^{(2^1)} & \lambda_2^{(2^1)} & \lambda_3^{(2^1)} & \lambda_4^{(2^1)} & \lambda_5^{(2^1)} & \lambda_6^{(2^1)} \\ \lambda_6^{(2^2)} & \lambda_7^{(2^2)} & \lambda_0^{(2^2)} & \lambda_1^{(2^2)} & \lambda_2^{(2^2)} & \lambda_3^{(2^2)} & \lambda_4^{(2^2)} & \lambda_5^{(2^2)} \\ \lambda_5^{(2^3)} & \lambda_6^{(2^3)} & \lambda_7^{(2^3)} & \lambda_0^{(2^3)} & \lambda_1^{(2^3)} & \lambda_2^{(2^3)} & \lambda_3^{(2^3)} & \lambda_4^{(2^3)} \\ \lambda_4^{(2^4)} & \lambda_5^{(2^4)} & \lambda_6^{(2^4)} & \lambda_7^{(2^4)} & \lambda_0^{(2^4)} & \lambda_1^{(2^4)} & \lambda_2^{(2^4)} & \lambda_3^{(2^4)} \\ \lambda_3^{(2^5)} & \lambda_4^{(2^5)} & \lambda_5^{(2^5)} & \lambda_6^{(2^5)} & \lambda_7^{(2^5)} & \lambda_0^{(2^5)} & \lambda_1^{(2^5)} & \lambda_2^{(2^5)} \\ \lambda_2^{(2^6)} & \lambda_3^{(2^6)} & \lambda_4^{(2^6)} & \lambda_5^{(2^6)} & \lambda_6^{(2^6)} & \lambda_7^{(2^6)} & \lambda_0^{(2^6)} & \lambda_1^{(2^6)} \\ \lambda_1^{(2^7)} & \lambda_2^{(2^7)} & \lambda_3^{(2^7)} & \lambda_4^{(2^7)} & \lambda_5^{(2^7)} & \lambda_6^{(2^7)} & \lambda_7^{(2^7)} & \lambda_0^{(2^7)} \end{pmatrix} \tag{4.2.9}$$

The seemingly arbitrary choices for the bottom 15 rows root in the concept of ϕ being a map mapping into *field conjugates*, a concept relating to minimal polynomials [52, p. 286], which we shall not dive into. Notice

$$(\hat{L}_B \phi(b))_0 = \sum_{k=0}^{7} \lambda_k b^{(2^k)} = f(b) \tag{4.2.10}$$

for any $b \in F$, which was the desired action. As for v_A, let $v_B \coloneqq \phi(v_A)$ and add it to the state byte. To apply the matrix on the entire state, it suffices to form the diagonal matrix composed of $4N_b$ L_B blocks and add v_B to each entry, while using S in a column-major enumeration.

(ii') `ShiftRows` [51, pp. 5-6]: The shifting of the F^8 elements inside S is the same as in the original AES. The matrix M_A is generalized to M_B by replacing every one with E_8 for a $F^{16N_b \times 16N_b}$ permutation matrix performing this action.

(iii') `MixColumns` [51, p. 6]: In the AES, we could represent this step as a matrix multiplication with M_A. In BES, we analogously define

$$C_B^k \coloneqq \begin{pmatrix} 2^{(2^k)} & 3^{(2^k)} & 1 & 1 \\ 1 & 2^{(2^k)} & 3^{(2^k)} & 1 \\ 1 & 1 & 2^{(2^k)} & 3^{(2^k)} \\ 3^{(2^k)} & 1 & 1 & 2^{(2^k)} \end{pmatrix} \in F^{4 \times 4} \tag{4.2.11}$$

for $k \in [0,7]_\mathbb{N}$. This matrix has the property

$$C_B^k \left(y_0^{(2^k)}, y_1^{(2^k)}, y_2^{(2^k)}, y_3^{(2^k)}\right)^t = \left(z_0^{(2^k)}, z_1^{(2^k)}, z_2^{(2^k)}, z_3^{(2^k)}\right)^t \tag{4.2.12}$$

for $y, z \in F^4$, preserving the aforementioned so-called conjugacy property. We may now set

$$M_B \coloneqq \begin{pmatrix} C_B^0 & \cdots & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \cdots & 0 & 0 & 0 \\ 0 & \cdots & C_B^7 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & C_B^0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \cdots & C_B^7 \end{pmatrix} \in F^{32N_b \times 32N_b} \tag{4.2.13}$$

Multiplying with S in column-major enumeration, while enumerating its entries from F^8 as row vectors gives the desired map.

(iv') `AddRoundKey` [51, p. 5]: We perform the same addition as in Rijndael $S \mapsto S + \tilde{K}_i$ in the $i \in [0, N_r]$th round.

(v') `KeySchedule` [51, p. 8]: All operations in the key schedule have been explained in the explanations of the previous suboperations, so we can carry it over identically by, instead of using F as the bytes in the key expansion array $W \in (F^8)^{4 \times 4N_b(N_r+1)} \cong F^{32 \times 4N_b(N_r+1)}$, the operations being addition and the map ρ, which takes on the form $b \mapsto \hat{L}_B(\iota(b_0), ..., \iota(b_7))^t + v_B$, and the polynomial addition. The round constants are also thus $\phi(r_j)$, $j \in \mathbb{N}_{\geq 1}$.

We may summarize this discussion with the following theorem.

Theorem 4.2. If $\alpha \colon F^{4 \times N_b} \times F^{4 \times N_k} \to F^{4 \times N_b}$ denotes the AES cipher and $\beta \colon F^{32 \times N_b} \times F^{32 \times N_k} \to F^{32 \times N_b}$ its associated BES-cipher respectively, then the following diagram commutes:

$$\begin{array}{ccc} F^{4 \times N_b} \times F^{4 \times N_k} & \xrightarrow{\phi_{4 \times N_b} \times \phi_{4 \times N_k}} & F^{32 \times N_b} \times F^{32 \times N_k} \\ \Big\downarrow{\scriptstyle \alpha} & & \Big\downarrow{\scriptstyle \beta} \\ F^{4 \times N_b} & \xleftarrow[\phi_{4 \times N_b}^{-1}]{} & F^{32 \times N_b} \end{array}$$

Note that in the above diagram, cartesian products of functions are functions taken component-wise and the arguments of α and β are in order (plaintext, key).

4.3 A BES Multivariate Equation System for AES

With the previous description of BES for AES, we can now form a multivariate equation system for key recovery over F. Note, that we do not follow Murphy and Robshaw with the derivation of their multivariate quadratic equation system in [51, pp. 11-13]. Consider the description of AES in Figure 15. We are given a plaintext $P \in (F^8)^{4 \times N_b}$ and a ciphertext $C \in (F^8)^{4 \times N_b}$, where we know, that C was produced from P via running BES using a key $K \in (F^8)^{4 \times N_k}$. We have an initial addition of the first $4 \times N_b$ sized part of the key K, which is $\tilde{K}_0$, giving the state S_0, then execute $N_r - 1$ "normal" rounds of AES before entering the last round, where `MixColumn` is omitted.

$$P \mapsto S_0 \mapsto S_1 \mapsto ... \mapsto S_{N_r-1} \mapsto S_{N_r} = C \tag{4.3.1}$$

This gives the following equation system over F.

$$\begin{array}{rcll} S_0 & = & P + \tilde{K}_0 & \\ T_i & = & L_B S_{i-1} + V_B & i \in [1, N_r - 1]_{\mathbb{N}} \\ U_i & = & S_B T_i & i \in [1, N_r - 1]_{\mathbb{N}} \\ S_i & = & M_B U_i + \tilde{K}_i & i \in [1, N_r - 1]_{\mathbb{N}} \\ T_{N_r} & = & L_B S_{N_r-1} + V_B & \\ U_{N_r} & = & S_B T_{N_r} & \\ S_{N_r} & = & U_{N_r} + \tilde{K}_{N_r} & \\ C & = & S_{N_r} & \end{array} \tag{4.3.2}$$

Here, $V_B = \begin{pmatrix} v_B & ... & v_B \end{pmatrix}^t \in (\mathbb{F}^8)^{4N_b}$. This is not the same system as in [51, pp. 11-13], where the addition of v_B and even the application of S_B were omitted. We further have the BES key schedule as

$$\begin{array}{rcll} w_{ij} & = & k_{ij} & i \in [0,3]_{\mathbb{N}}, j \in [0, N_k - 1]_{\mathbb{N}} \\ w_{0j} & = & w_{0(j-N_k)} + L_B w_{1(j-1)} + V_B + r_{j/N_k} & j \in [N_k, N_b(N_r+1) - 1]_{\mathbb{N}}, j = 0 \bmod N_k \\ w_{ij} & = & w_{i(j-N_k)} + L_B w_{((i+1) \bmod 4)(j-1)} + V_B & i \in [1,3]_{\mathbb{N}}, j \in [N_k, N_b(N_r+1) - 1]_{\mathbb{N}}, j = 0 \bmod N_k \\ w_{ij} & = & w_{i(j-N_k)} + L_B w_{i(j-1)} + V_B & i \in [0,3]_{\mathbb{N}}, j \in [N_k, N_b(N_r+1) - 1]_{\mathbb{N}}, \\ & & & j = 4 \bmod N_k, N_r > 6 \\ w_{ij} & = & w_{i(j-N_k)} + w_{i(j-1)} & i \in [0,3]_{\mathbb{N}}, j \in [N_k, N_b(N_r+1) - 1]_{\mathbb{N}}, j \neq 0 \bmod N_k \\ \tilde{K}_i & = & (w_{i'j'})_{i' \in [0,3]_{\mathbb{N}}, j' \in [N_b i, N_b(i+1)-1]_{\mathbb{N}}} & \end{array} \tag{4.3.3}$$

The last aspect we want to analyze wrt. the BES system is the size of this system, and thus the variable and equation count. Consider first the initial equation system. From the count, we omit the variables $U_1, ..., U_{N_r}$, as they are just permutations of the existing $T_1, ..., T_{N_r}$ variables. We also do not count the variables for W in the key schedule. Furthermore, we omit the equations for the U_i's, $C = S_{N_r}$ and for the $\tilde{K}_i$ assignments. Consider the following table, in which we count the number of variables and equations in each line, including the key schedule, where we mean by "new", that the variables appearing in the equation have not appeared in a previous row.

Equation	New Variables	Equations	Occurences
$S_0 = P + \tilde{K}_0$	$8N_b$	$32N_b$	1
$T_i = L_B S_{i-1} + V_B$	$4N_b$	$32N_b$	$N_r - 1$
$U_i = S_B T_i$	Omitted	Omitted	Omitted
$S_i = M_B U_i + \tilde{K}_i$	$8N_b$	$32N_b$	$N_r - 1$
$T_{N_r} = L_B S_{N_r - 1} + V_B$	$4N_b$	$32N_b$	1
$U_{N_r} = S_B T_{N_r}$	Omitted	Omitted	Omitted
$S_{N_r} = U_{N_r} + \tilde{K}_{N_r}$	$8N_b$	$32N_b$	1
$C = S_{N_r}$	Omitted	Omitted	Omitted
$w_{ij} = k_{ij}$	Omitted	Omitted	Omitted
$w_{0j} = w_{0(j-N_k)} + L_B w_{1(j-1)} + V_B + r_{j/N_k}$	Omitted	8	≤ 15
$w_{ij} = w_{i(j-N_k)} + L_B w_{((i+1) \bmod 4)(j-1)} + V_B$	Omitted	8	≤ 45
$w_{ij} = w_{i(j-N_k)} + L_B w_{i(j-1)} + V_B$	Omitted	8	≤ 60
$w_{ij} = w_{i(j-N_k)} + w_{i(j-1)}$	Omitted	8	≤ 180
$\tilde{K}_i = (w_{i'j'})_{i' \in [0,3]_{\mathbb{N}}, j' \in [N_b i, N_b(i+1)-1]_{\mathbb{N}}}$	Omitted	Omitted	Omitted

Table 2: Sizes of equations in the BES system, where we upper bound the occurences of some of the key schedule equations by letting $(N_k, N_r) = (4, 14)$ wlog..

Theorem 4.3 (Equation System Size for BES Key Recovery)**.** Using BES, the key for a given AES encryption can be recovered using an equation system of

$$20N_b + (N_r - 1)12N_b \text{ variables and } 96N_b + (N_r - 1)64N_b + 2400 \text{ equations.} \tag{4.3.4}$$

(N_k, N_r)	$(4, 10)$	$(6, 12)$	$(8, 14)$
(m, n)	$(416, 4576)$	$(512, 5088)$	$(608, 5600)$

Table 3: Direct BES system sizes. $m \in \mathbb{N}$ is the variable count and $n \in \mathbb{N}$ the equation count each. $N_b = 4$ for AES, as previously said. These systems are not yet linearized.

The aforementioned construction for a multivariate equation system for AES using BES demonstrates the technique. Consider also, that we, with this construction, have a system, where the polynomial degrees range up to 128, as we have directly used the conjugates in the system. The following subsection is dedicated to presenting recent results for solving this system of equations.

4.4 Overview of Recent Research on the Approach

We present a discussion of recent results on the cryptanalysis of AES, especially under the HHL algorithm, by studying the results of three research groups by Courtois, Chen and Ding.

Algebraic Cryptanalysis via XSL

The classical literature on the cryptanalysis of AES is extensive [49, 50, 53, 54]. In this paragraph, we focus on algebraic cryptanalysis using linear systems of equations, as we have been aluding to. We present three major results, along with the previous results by Murphy and Robshaw.
In 2002, the cryptanalysts Courtois and Pieprzyk presented the so-called *Extended Sparse Linearization* (XSL) attack on block ciphers, especially on AES [55]. It improved upon the previous *Extended Linearization* (XL) technique. The essential idea is to form systems of *multivariate quadratic* (MQ) equations, which are then formed into SLEs by introducing variables for the monomials [53, p. 2]. XSL attempts to utilize the case, where that equation system is massively overdefined. XSL came under quite some controversy, especially since the effectiveness of the attacks is largely debated [53, p. 2] [50, pp. 15-16]. XSL has to this day never been implemented.
The original XSL paper is also not very explicit wrt. the actual construction of the MQ system. Courtois and Pieprzyk claim the following result.

Theorem 4.4 (Direct Rijndael MQ System Complexity)**.** The problem of recovering the key from a Rijndael encryption of one plaintext with parameters (N_b, N_k, N_r) can be reduced to the problem of solving an MQ system with m quadratic equations and n variables, where

$$m = 160N_bN_r + 5(L_k - 32N_k) \qquad n = 32N_b(N_r - 1) + L_k \tag{4.4.1}$$

with

$$L_k \coloneqq \begin{cases} 32\left(N_k + \left\lceil \frac{N_bN_r+N_b-N_k}{N_k} \right\rceil\right) & N_k \neq 8 \\ 32\left(N_k + \left\lceil \frac{N_bN_r+N_b-N_k}{4} \right\rceil\right) & N_k = 8 \end{cases} \tag{4.4.2}$$

The constants in the theorem are a direct result of using the theorem on [55, p. 22] and substituting $r \coloneqq 40$ as on the same page and $s = 8$ from p. 4. Note, that we have $H_k = 32N_k$ in their description, following pp. 3-4. The definition for L_k can be found on p. 21.

(N_k, N_r)	$(4, 10)$	$(6, 12)$	$(8, 14)$
(m, n)	$(8000, 1600)$	$(9600, 1920)$	$(11200, 2240)$

Table 4: Direct AES MQ system sizes. $N_b = 4$ for AES, as previously said. These systems are not yet linearized.

The equation systems of Murphy et al. and Courtois et al. are not the same. The techniques proposed by Courtois et al. yield the following theorem, see [55, p. 13], which we shall not further study.

Theorem 4.5. Performing an XSL attack on AES-128 requires approximately

$$T^\omega \approx 2^{230} \tag{4.4.3}$$

operations classically.

XSL is one of the more widely known approaches to the algebraic cryptanalysis of AES. Few authors have yet considered using the HHL algorithm for this task. We briefly overview the results by Chen and Gao [4, 54] and, following their results, Ding et al. [5].

Chen and Gaos Results

Chen and Gao investigated the applicability of HHL on the cryptanalysis of AES in a longer 2017 paper [4]. The essential idea of using a linear system for the cryptanalysis is not considered at first, but rather the problem of solving Boolean polynomial equation systems directly. The HHL algorithm itself poses three challenges to this problem:

(i) The algorithm yields a result vector over the field $\mathbb{C}$ and not GF(2). This can be mitigated by including additional equations of form $\{x_1^2 - x_1, ..., x_n^2 - x_n\}$ with $x_1, ..., x_n$ being the variables inside of the original Boolean polynomial system, as in $\mathbb{C}$ each equation can only be satisfied, iff $x_1, ..., x_n \in \text{GF}(2)$ using $x_i = |x_i|e^{i \arg(x_i)}$ for $i \in [1, n]_{\mathbb{N}}$.

(ii) HHL may produce a wrong result or it may produce a result despite the system being unsolvable.

(iii) The result is a quantum state and not a classically accessible Boolean vector.

An Application of HHL The first major result is the application of the HHL algorithm under two assumptions [4, pp. 6-8].

I. The given matrix $A \in \mathbb{C}^{M \times N}$, $M \coloneqq r2^{\nu}$ with $r, \nu \in \mathbb{N}_{\geq 1}$, $N \in \mathbb{N}_{\geq 1}$ is s-sparse and possesses a decomposition into s 1-sparse matrices of form $A = \sum_{j=1}^{s} A_j$, where the entries of each matrix A may be queried in time $O(\gamma)$ with γ being a complexity term.

II. The given vector $b \in \{0, 1\}^M$ suffices $b_i = 1$, iff $i = k2^{\nu}$ for $k \in [0, \rho - 1]_{\mathbb{N}}$ for a $\rho \in [0, r]_{\mathbb{N}}$.

We may especially note the very tiny decomposition of A into s other matrices. Chen and Gao describe the effects of the assumptions and the decomposition on the algorithm runtime, as well as the efficient initializability of the state $|b\rangle$, which is associated to b. The result is then, that

Theorem 4.6. Given the matrix A and the vector b as in the stated assumptions and under the use of the HHL algorithm as in Algorithm 2, as well as an error cap $\varepsilon \in \mathbb{R}_{>0}$, the linear system of equations $Ax = b$ can be solved in time $\tilde{\mathcal{O}}((\log(M + N) + \gamma)s\kappa^2/\varepsilon)$.

A Sufficiently Sparse Macaulay System for Boolean Polynomial Equation Systems For a given multivariate Boolean polynomial equation system $\mathcal{F} \coloneqq \{f_1, ..., f_r\} \subseteq \text{GF}(2)[x_1, ..., x_n]$, $r \in \mathbb{N}_{\geq 1}$, $n \in \mathbb{N}_{\geq 1}$ to be solved, meaning, that we want to find some $s \in \text{GF}(2)^n$ with $f_1(s) = ... = f_r(s) = 0$, Chen and Gao develop a *Macaulay linear system*, i.e. an SLE describing the structure of a polynomial equation system, which suffice the assumptions stated in Section 4.4. The construction of the Macaulay linear system involves a bit of machinery, so we may omit it. It can be found on [4, pp. 8-11]. The next theorem summarizes the result.

Theorem 4.7. Let $T_{\mathcal{F}} \coloneqq \sum_{f \in \mathcal{F}} t_f$ be the so-called *total sparseness* of $\mathcal{F}$, where $t_f \in \mathbb{N}$ denotes the number of terms in a given Boolean polynomial f. A given polynomial Boolean equation system $\mathcal{F}$ can be described by a Macaulay linear system $M_{\mathcal{F}} x = b_{\mathcal{F}}$ with the following properties:

a) $M_{\mathcal{F}}$ is $T_{\mathcal{F}}$-sparse and $M_{\mathcal{F}}$ can be decomposed into $T_{\mathcal{F}}$ 1-sparse matrices, each of which may be queried in time $\mathcal{O}(n \log_2(D) + \log_2(r))$ for some $D \in \mathbb{N}$, s.t. $D \geq \max_{f \in \mathcal{F}} d_f$ with $d_f \in \mathbb{N}$ being the total degree of f, i.e. the maximum of the sums of the degrees in each monomial. So $M_{\mathcal{F}}$ suffices assumption I.

b) $b_{\mathcal{F}}$ suffices assumption II.

Resulting Algorithms Using the previous two results, the authors describe multiple algorithms. First, we consider a general algorithm for solving a given multivariate Boolean equation system by solving it over $\mathbb{C}[x_1, ..., x_n]$ first using a quantum algorithm. We call a solution to such a complex system *boolean*, if all of the entries in the result vector are in GF(2).

Theorem 4.8. Given a polynomial equation system $\mathcal{F} \subseteq \mathbb{C}[x_1, ..., x_n]$ and an error cap $\varepsilon \in \mathbb{R}_{>0}$, there is a quantum algorithm, which decides the solvability of $\mathcal{F}$ for recovering a Boolean solution, i.e. one in $\mathbb{F}_2^n$, with success probability at least $1 - \varepsilon$ and, if so, returns a solution vector in time

$$\tilde{\mathcal{O}}(n^{2.5}(n + T_{\mathcal{F}})\kappa^2 \log_2(1/\varepsilon)) \tag{4.4.4}$$

where κ denotes the maximal condition number of the linear system fpr $\mathcal{F}'_B \cup \{x_1^2 - x_1, ..., x_n^2 - x_n\}$ with $\mathcal{F}'_B$ being the, during the algorithms execution, modified system, where any occurence of x_i^m has been replaced with x_i for $i \in [1, n]_{\mathbb{N}}$ and $m \in \mathbb{N}$.

The description and proof of runtime can be found on [4, pp. 16-19]. We further have, in the same manner as Grovers algorithm, a result regarding multiple solutions.

Theorem 4.9. The quantum algorithm described in Theorem 4.8 can be extended to find all $\omega \in \mathbb{N}$ solutions in time

$$\tilde{\mathcal{O}}(n^{2.5}(n + T_{\mathcal{F}} + \omega)\omega\kappa^2 \log_2(1/\varepsilon)) \tag{4.4.5}$$

with success probability $(1 - \varepsilon)^{\omega}$.

These two results are further modified for linear Boolean systems, which we will not further present.

Application to AES Lastly, the results for solving Boolean polynomial systems are applied to AES using the BES cipher, see Section 4.2, in [4, pp. 24-25, pp. 32-34]. The obtained result is summarized in the following theorem.

Theorem 4.10. There exists a quantum algorithm, which recovers the key of an AES encryption in time

$$\begin{cases} O(\sqrt{2}\alpha_0\alpha_1^{2.5}\alpha_2\kappa^2 \log_2(1/\varepsilon)) & N_k \leq 6 \\ O(\sqrt{2}\beta_0\beta_1^{2.5}\beta_2\kappa^2 \log_2(1/\varepsilon)) & N_k > 6 \end{cases} \tag{4.4.6}$$

with

$$\begin{pmatrix} \alpha_0 & \beta_0 \\ \alpha_1 & \beta_1 \\ \alpha_2 & \beta_2 \end{pmatrix} := \begin{pmatrix} \log_2(5024N_kN_r + 224N_k + 5472N_r) + 3 & \log_2(5024N_kN_r + 224N_k + 10272N_r) + 3 \\ 5024N_kN_r + 224N_k + 5472N_r & 5024N_kN_r + 224N_k + 10272N_r \\ 34592N_kN_r + 1376N_k + 38112N_r & 34592N_kN_r + 1376N_k + 71520N_r \end{pmatrix} \tag{4.4.7}$$

To illustrate the runtimes, consider the following table.

AES-version	Runtime Factor
AES-128	$2^{73.30}$
AES-192	$2^{76.69}$
AES-256	$2^{78.53}$

Table 5: Runtimes of the AES key-recovery algorithm proposed by Chen and Gao, taken directly from [4, p. 26]. The runtime factor is without any asymptotic factors or the squared condition number.

Discussion We may be sceptical of the results presented, especially in the equation system used for the key recovery for AES. For one, it is not clear, why the presented system corresponds to the BES system described by Murphy and Robshaw. Also, the time of the Hamiltonian simulation may be wrong due to a mistake by Harrow, as we have argued in Section 3.4. Chen and Gao have called their algorithm "complicated" [4, p. 5], possibly partly because of the rather sophisticated Gröbner basis techniques used in [4, pp. 11-15], [5, p. 2] have given a more elementary proof in their improved version of the algorithm. There have also been criticisms voiced by other researchers. For instance, Gao et al. argues [54, p. 2], that the equation system for AES is incomplete. Furthermore, a major question was left open, which is the range of the condition number κ. However, what the result indicates, is a first hint to the HHL algorithm not being sufficient for the cryptanalysis of AES, which is be further supported by the next paragraph.

Further Research by Ding et al.

Bounding the condition number is essential to obtain a clear bound on the runtime of Chen and Gaos cryptanalysis. Ding et al. prove a lower bound on the condition number, which depends on the sparsity of the solution vector of a linear system [5].

Two Preliminary Notions We recall and introduce a few notions. One essential concept here is the *truncated condition number*.

Definition 4.11. Let $A \in \mathbb{C}^{m \times n}$ with $m, n \in \mathbb{N}_{\geq 1}$ and $b \in \mathbb{C}^n \setminus \{0\}$. The *truncated condition number* $\kappa_b(A)$ of the linear system $Ax = b$ is defined as

$$\kappa_b(A) \coloneqq \|A\| \frac{\|A^+ b\|}{\|b\|} \tag{4.4.8}$$

where the norm used is the operator norm and A^+ is the Moore-Penrose Pseudoinverse of A following Definition 1.15.

Lemma 4.12. For any matrix A and vector b as in Definition 4.11, we have

$$\kappa_b(A) \leq \kappa(A) \tag{4.4.9}$$

Proof. Holds by

$$\left\|A^+\right\| \geq \left\|A^+ b\right\| / \|b\| \tag{4.4.10}$$

■

The importance of this lemma lies in the fact, that lower bounds for condition numbers can be acquired with truncated condition numbers.

Theorem 4.13. For binary vectors $u, v \in \{0,1\}^n$ define the *Hamming distance* and *Hamming weight* as

$$d_H \colon \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{N}, (u, v) \mapsto |\{i \mid i \in [1, n]_{\mathbb{N}} \wedge u_i \neq v_i\}| \qquad w_H \colon \mathbb{F}_2^n \to \mathbb{R}_{\geq 0}, u \mapsto \sqrt{d_H(u, 0)} \tag{4.4.11}$$

Then d_H is a metric and w_H is a norm.

These terms are known from general coding theory, see [56, pp. 100-101].

Lower Bounds of Truncated Condition Numbers for Macaulay Linear Systems As we have seen, lower bounding a fixed truncated condition number is sufficient for lower bounding the runtime of Chen and Gaos procedures. The results regarding the truncated condition number in the context of the Macaulay matrices involved in Chen and Gaos research can be found in [5, pp. 8-13]. Whilst we have not introduced the Macaulay matrix or even the matrices involved in the original work, we shall still present the results.
For a given Macaulay SLE problem $\mathcal{M}x = b$ with $\mathcal{M}$ denoting the Chen and Gao Macaulay system for key recovery and following [5, pp. 8-9], we have

$$\|\mathcal{M}\| \geq 1 \tag{4.4.12}$$

Thus, if we assume wlog., as described in Section 3.4, $\|b\| = 1$, we have

$$\kappa(\mathcal{M}) \geq \kappa_{|b\rangle}(\mathcal{M}) \geq \|y\| \tag{4.4.13}$$

where y denotes a solution vector to the system. The reduction that follows in [5, pp. 9-10], in combination with a few elementary results regarding sets in binary vector spaces yield the following theorem.

Theorem 4.14. Let $\mathcal{F} \subseteq \mathbb{C}[x_1, ..., x_n]$, $n \in \mathbb{N}_{\geq 1}$, be a polynomial equation system and $\mathcal{M}x = b$ be the Macaulay system for finding a Boolean solution to $\mathcal{F}$ proposed by Chen and Gao. Then we have for the $t \in \mathbb{N}_{\geq 1}$ solutions of $\mathcal{F}$ $a_1, ..., a_t \in \mathbb{F}_2^n$

$$w_H(a_1) = ... = w_H(a_t) \tag{4.4.14}$$

and

$$\kappa_b(\mathcal{M}) \geq \sqrt{(3n)^h / t} \tag{4.4.15}$$

where $h \coloneqq w_H(a_1)$.

In [5, pp. 13-21], Ding et al. have further presented an improved Macaulay system and associated solution algorithm.

Remark 4.15 (The Condition Number of the Macaulay Matrix as a Block Cipher Design Criterion). The example of the algebraic cryptanalysis of AES demonstrates the possibility of formulating block cipher key recovery problems as large equation systems. Following the results we have discussed, especially by Ding et al., we may draw two conclusions to the design criterions a block cipher should fulfill.
First, it seems tempting to conclude, that a huge number of dependencies yielding a large size for a key recovery equation system may suffice to make a cipher strong. But one major point in the design of XSL was, that if the system is huge, but massively overdefined, an attacker may be able to break it very fast [55, p. 15].
Second, given the results presented, a designer of a block cipher should apply the results by Chen and Gao and Ding et al. on the key recovery equation systems, which are associated to the cipher. While Chen and Gao give the quantum algorithm for breaking it, the results by Ding et al. show the impossibility of key recovery using this technique, if the associated bound is very low. The general exponential lower bound also yields some content for discussions: Is the Macaulay approach in this form not sufficient and can be substantially improved or does the analysis by Ding et al. at some point go wrong? If neither case holds, there is one more possibility: Inside of the cryptographic community, there has been discussion on whether the condition number of associated key recovery equation systems of a block cipher fulfill more general design criteria for block ciphers. We are not aware of recent major results in this topic. Such a result would most likely make this approach unfeasible in general.

A Omitted Details

Lemma 2.22. For any $\alpha \in \mathbb{C}$ and $m \in \mathbb{N}_{\geq 1}$, we have

$$\sum_{j=0}^{m-1} \cos((2j+1)\alpha) = \frac{\sin(2m\alpha)}{2\sin(\alpha)} \tag{2.5.22}$$

Proof by induction over m. For $m = 1$, consider

$$\cos(\alpha) = \frac{2\cos(\alpha)\sin(\alpha)}{2\sin(\alpha)} = \frac{\sin(2\alpha)}{2\sin(\alpha)} \tag{A.0.1}$$

under the use of Theorem B.4.
Suppose the statement holds for an arbitrary, but fixed m. Then for the inductive step, under the usage of the assumption, the addition of a skillful zero and using Theorem B.4 twice, we obtain

$$\sum_{j=0}^{(m+1)-1} \cos((2j+1)\alpha) = \frac{\sin(2m\alpha)}{2\sin(\alpha)} + \cos((2m+1)\alpha) \tag{A.0.2}$$

$$= \frac{\sin((2m+1)\alpha - \alpha) + 2\cos((2m+1)\alpha)\sin(\alpha)}{2\sin(\alpha)} \tag{A.0.3}$$

$$= \frac{-\cos((2m+1)\alpha)\sin(\alpha) + \sin((2m+1)\alpha)\cos(\alpha) + 2\cos((2m+1)\alpha)\sin(\alpha)}{2\sin(\alpha)} \tag{A.0.4}$$

$$= \frac{\sin(2(m+1)\alpha)}{2\sin(\alpha)} \tag{A.0.5}$$

By the principle of the theorem of induction, the statement is proven. ■

Lemma 3.11. The following bounds hold for any $x \in \mathbb{R}_{\geq 0}$:

$$x - \frac{x^3}{6} \leq \sin(x) \leq x \tag{3.3.11}$$

With strict inequalities for $x \neq 0$.

Proof. Let $f\colon \mathbb{R} \to \mathbb{R}, x \mapsto x - \frac{x^3}{6}$ and let $x \in \mathbb{R}_{\geq 0}$ be fixed. From real analysis we know that $f'(x) = 1 - \frac{x^2}{2}$ and thus that f is monotonically decreasing in $[\sqrt{2}, \infty)$. Especially $f(3) = -\frac{3}{2}$. So the first inequality holds in $[3, \infty)$. We prove the statement for $[0, 3)$.
The sine can be represented as a sum of some first terms of its Taylor series Theorem B.1, and in sum with the following Langrangian remainder terms for some $\xi_1, \xi_2 \in [0, x]$:

$$\sin(x) = x - \frac{x^3}{6} + \frac{\sin(\xi_1)}{4!}x^4 = x - \frac{\sin(\xi_2)}{2!}x^2 \tag{A.0.6}$$

See [38, p. 284]. Since $\sin(\xi_2) \geq 0$ for $x \in [0, 1)$, we have the upper bound. For $x \in [0, \pi]$ and especially $x \in [0, 3)$ we have $\sin(\xi_1) \geq 0$, thus concluding the lower bound. The strict inequality can be read off directly. ■

Lemma 3.15. Defining for $T \coloneqq 2^t$, $t \in \mathbb{N}_{\geq 5}$

$$l^{\uparrow}\colon [2\pi, \pi T] \to \mathbb{R}, \delta \mapsto \sin\left(\frac{\delta + \pi}{2T}\right)\sin\left(\frac{\delta - \pi}{2T}\right) \qquad l^{\downarrow}\colon [2\pi, \pi T] \to \mathbb{R}, \delta \mapsto \frac{c_1}{\pi^2}\frac{\delta^2}{T^2} \tag{3.3.32}$$

where $c_1 \coloneqq 0.9975 < \sin\left(\frac{\pi}{2} - \frac{\pi}{64}\right)$, we have $l^{\uparrow} > l^{\downarrow}$.

Proof. We first calculate the derivatives of both functions (i) and then divide the interval into two pieces, for which we argue the statement analytically (ii) and geometrically (iii).

(i) Observe, that

$$l^{\uparrow\prime}(\delta) = \frac{1}{2T}\left(\cos\left(\frac{\delta+\pi}{2T}\right)\sin\left(\frac{\delta-\pi}{2T}\right) + \sin\left(\frac{\delta+\pi}{2T}\right)\cos\left(\frac{\delta-\pi}{2T}\right)\right) \overset{(1)}{=} \frac{1}{2T}\sin\left(\frac{\delta}{T}\right) \tag{A.0.7}$$

$$l^{\downarrow\prime}(\delta) = \frac{2c_1}{\pi^2}\frac{\delta}{T^2} \tag{A.0.8}$$

(1) We use Theorem B.4.

Notice, that both functions grow strictly monotonically. Consider the partition

$$[2\pi, \pi T] = \left[2\pi, \frac{\pi}{2}T\right] \cup \left(\frac{\pi}{2}T, \pi T\right] \tag{A.0.9}$$

(ii) It suffices to show, that $l^{\uparrow\prime} > l^{\downarrow\prime}$ and $\left(l^{\uparrow}(2\pi), l^{\uparrow}\left(\frac{\pi}{2}T\right)\right) > \left(l^{\downarrow}(2\pi), l^{\downarrow}\left(\frac{\pi}{2}T\right)\right)$. Let $\delta \in \left[2\pi, \frac{\pi}{2}T\right]$. Going from left to right, we first have with Lemma 3.11 and $-\delta^2 \geq -\frac{\pi^2}{4}T^2$

$$l^{\uparrow\prime}(\delta) > \frac{1}{2T}\frac{\delta}{T}\left(1 - \frac{1}{6}\frac{\delta^2}{T^2}\right) \geq \frac{1}{2}\left(1 - \frac{\pi^2}{24}\right)\frac{\delta}{T^2} > \frac{2c_1}{\pi^2}\frac{\delta}{T^2} = l^{\downarrow\prime}(\delta) \tag{A.0.10}$$

Then, it holds, that

$$l^{\uparrow}(2\pi) > \frac{3\pi^2}{4T^2}\left(1 - \frac{1}{6}\frac{10\pi^2}{4T^2}\right) > \frac{c_3}{T^2} > \frac{4c_1}{T^2} = l^{\downarrow}(2\pi) \tag{A.0.11}$$

where $c_3 \coloneqq 7.3724 < \frac{3\pi^2}{4}\left(1 - \frac{1}{6}\frac{10\pi^2}{4\cdot 32^2}\right)$. Note, that we use Lemma 3.11 twice in the product for the first lower bound, which is allowed, as the argument is still inside of $(0, \pi/2)$. The third claim follows directly from the larger strictly monotonic growth of $l^{\uparrow}$ and the initial inequality at 2π.

(iii) Since $l^{\uparrow\prime\prime}\Big|_{\left(\frac{\pi}{2}T, \pi T\right]} < 0$, the function is concave [38, pp. 185-187], whilst the parabola $l^{\downarrow}$ is clearly convex. First, we have with the sinoal symmetry around $\frac{\pi}{2}T$

$$l^{\uparrow}(\pi T) = \sin^2\left(\frac{\pi}{2} - \frac{\pi}{64}\right) > c_1 = l^{\downarrow}(\pi T) \tag{A.0.12}$$

We have $g^{\uparrow} > g^{\downarrow}$, where $g^{\uparrow}$ is the line segment connecting $\left(\frac{\pi}{2}T, l^{\uparrow}\left(\frac{\pi}{2}T\right)\right)$ and $(\pi T, l^{\uparrow}(\pi T))$, and where $g^{\downarrow}$ connects $\left(\frac{\pi}{2}T, l^{\downarrow}\left(\frac{\pi}{2}T\right)\right)$ and $(\pi T, l^{\downarrow}(\pi T))$ respectively. We have

$$l^{\uparrow}\big|_{\left(\frac{\pi}{2}T, \pi T\right]} \geq g^{\uparrow} > g^{\downarrow} \geq l^{\downarrow}\big|_{\left(\frac{\pi}{2}T, \pi T\right]} \tag{A.0.13}$$

concluding the proof.

■

B Formula Sheet

This appendix presents some of the formulas used. We refer to [38] and [57], but any undergraduate Analysis and Complex Analysis textbook will most likely present these results.

Theorem B.1 (Exponential, Sine and Cosine Taylor Series)**.** For any $x \in \mathbb{C}$ it holds that:

$$\exp(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!} \qquad \sin(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!} \qquad \cos(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!} \tag{B.0.1}$$

[38, p. 288], gives a detailed calculation of the latter two expansions for $\mathbb{R}$. [57, p. 5], states these power series expansions for $\mathbb{C}$.

Definition B.2 (Exponential Sine and Cosine)**.** For any $x \in \mathbb{C}$ we have:

$$\sin(x) \coloneqq \frac{e^{ix} - e^{-ix}}{2i} \qquad \cos(x) \coloneqq \frac{e^{ix} + e^{-ix}}{2} \tag{B.0.2}$$

The statement for $\mathbb{R}$ can be found in [38, pp. 146-147], it is clear that both can be obtained by direct calculation and also hold for $\mathbb{C}$.

Theorem B.3 (Trigonometric Pythagoras)**.** For any $x \in \mathbb{C}$ the following holds:

$$\sin^2(x) + \cos^2(x) = 1 \tag{B.0.3}$$

The proof can be found in [38, p. 140].

Theorem B.4 (Sine and Cosine Addition Theorems)**.** For any $x, y \in \mathbb{C}$ it holds that:

$$\sin(x+y) = \sin(x)\cos(y) + \cos(x)\sin(y) \qquad \cos(x+y) = \cos(x)\cos(y) - \sin(x)\sin(y) \tag{B.0.4}$$

Again, the proof can be found in [38, p. 140].

Theorem B.5 (Geometric Sum)**.** For any $q \in \mathbb{C}$ and $n \in \mathbb{N}$, we have

$$\sum_{i=0}^{n} q^i = \begin{cases} \frac{1-q^{n+1}}{1-q} & q \neq 1 \\ n+1 & q = 1 \end{cases} \tag{B.0.5}$$

Proof. The first case follows directly from $\left(\sum_{i=0}^{n} q^i\right)(1-q) = 1 + \left(\sum_{i=1}^{n} \left(-q^i + q^i\right)\right) - q^{n+1}$, the second case by addition. ∎

Theorem B.6 (Cauchy-Schwarz Inequality)**.** For any $x, y \in \mathbb{C}^n$, $n \in \mathbb{N}_{\geq 1}$, we have

$$|\langle x, y \rangle| \leq \|x\| \|y\| \tag{B.0.6}$$

The proof can be found in [2, p. 220].

Theorem B.7. We have

$$\sum_{k=1}^{\infty} \frac{1}{k^4} = \frac{\pi^4}{90} \tag{B.0.7}$$

In close relation to the Riemann ζ-function, the proof of this limit can be found in [58, pp. 296-298].

C Hardness Results

In this appendix, we quickly present the hardness results on the solving of SLEs via quantum algorithms by Harrow et al. [3, pp. 12-14]. This topic is not included in the main body of the thesis, but shall be visited, s.t. we have worked through the entire original HHL paper, as well as got some intuition on the computational complexity theory of matrix inversion.

PSPACE, PP, BPP and BQP

The landscape of computational complexity classes is vast. Besides classes dedicated to capturing the time complexity of a problem, there are also notions for considering the space complexity of a problem. We use the books by Sipser and Barak [18, 59]. Recall the concepts of a language [59, p. 16], computability in the sense of Turing machines [59, p. 168], the complexity classes P [59, p. 286] and NP [59, pp. 293-294], SAT [59, p. 299] and polynomial time reducibility [59, p. 300]. We do not introduce these classes rigorously. Let Σ be an alphabet.

Definition C.1. We define the following notions.

(i) The complexity class PSPACE is composed of all languages $L \subseteq \Sigma^*$, for which the associated decision problem $\Sigma^* \ni \omega \in L$ can be decided with polynomial space complexity.

(ii) A language $B \subseteq \Sigma^*$ is called PSPACE-complete, if $B \in$ PSPACE and $A \leq_p B$ for any $A \in$ PSPACE.

(iii) The language TQBF is composed of all Boolean formulas with existential or universal quantifiers, for which an assignment of the quantifiers making the associated statement true exists.

These definitions follow [59, pp. 336-338], where we cite in this order.

Theorem C.2. TQBF is PSPACE-complete.

The proof is based on a Savitchs Theorem and can be found in [59, pp. 339-341].

Besides P, NP and PSPACE, classes dedicated to decision problems, other classes have arisen. We want to consider probabilistic and especially quantum complexity classes. The following definition captures the most important classes we want to know about.

Definition C.3. We define the following complexity classes.

(i) The complexity class PP is the set of languages $L \in \Sigma^*$, s.t. there is a probabilistic polynomial-time Turing machine T, s.t. it successfully and a polynomial $p \in \mathbb{R}[x]$, $p(\mathbb{N}) \subseteq \mathbb{N}$, s.t. $x \in L$, $|x| \in \mathbb{N}$ denoting the word length, iff

$$\left|\left\{y \in \Sigma^{p(|x|)} \mid M(x,y) = 1\right\}\right| \geq \frac{1}{2} 2^{p(|x|)} \tag{C.0.1}$$

(ii) The complexity class BPP is the set of languages $L \in \Sigma^*$, s.t. there is a probabilistic polynomial-time Turing machine T, s.t. it successfully and a polynomial $p \in \mathbb{R}[x]$, $p(\mathbb{N}) \subseteq \mathbb{N}$, s.t. $x \in L$, $|x| \in \mathbb{N}$ denoting the word length, iff

$$\left|\left\{y \in \Sigma^{p(|x|)} \mid M(x,y) = 1\right\}\right| \geq \frac{2}{3} 2^{p(|x|)} \tag{C.0.2}$$

(iii) The complexity class BQP is the set of languages $L \in \Sigma^*$, s.t. there is a polynomial-time quantum algorithm, which solves the decision problem $w \in L$ for a $w \in \Sigma^*$ with a success probability of at least 2/3.

The notion of completeness carries over analogously from PSPACE-completeness. We interpret (i) of this definition by considering x to be the problem instance of interest and y to be one of the possible solutions. The polynomial p thus computes the required length of y. The bound on the right specifies, that at least half of these possible values of y are valid solutions to the problem given by x. Another

way of phrasing this is, that there is an algorithm, specified by M and p, which solves $x \in L$ with a probability of at least one half, as we may then choose y uniformly at random. These first two definitions, with this equivalent interpretation, follow the book from Arora and Barak [18, p. 173, pp. 116-117]. The definition of BQP follows [18, p. 412], although we do not directly fix the set of allowed unitaries.

The following three results are known.

Theorem C.4. The following statements hold.

(i) BQP $\subseteq$ PSPACE.

(ii) BPP $\subseteq$ BQP.

(iii) BQP $\subseteq$ PP.

The first and third parts are major results from [7, p. 201] and [60, p. 1538]. Both proofs utilize a technique called *sum of histories*. The second statement follows from the fact, that we can simulate any classical simulation using a gate quantum algorithm, so also probabilistic algorithms, as Nielsen and Chuang point out [7, p. 201]. It is interesting to note, that all converse directions are unknown, but provide possibilities for determining the hardness of a problem in a quantum complexity class. Especially, the second statement is widely conjectured to be false.

The Problem MATRIXINVERSION

Following Harrow et al., we give a formal definition for the problem of matrix inversion. Furthermore, we present the hardness results from the HHL paper and explain their hardness with regard to the previously presented complexity classes and problems.

Definition C.5. Let $N \coloneqq 2^n$ for $n \in \mathbb{N}_{\geq 2}$. We say, that a quantum algorithm solves the problem MATRIXINVERSION, if for a given Hermitian, $O(1)$-sparse matrix $A \in \mathbb{C}^{N \times N}$ with $\kappa \coloneqq \kappa(A)$ and $\frac{1}{\kappa} \leq \lambda \leq 1$ for any eigenvalue $\lambda \in \mathbb{R}$ of A, for which the entries in a row can either be computed by an algorithm with runtime $\text{poly}(\log_2(N))$ or an oracle, it computes a quantum state $|x\rangle$ with $\left\| |x\rangle - \frac{1}{\|A^{-1}|0\rangle\|} A^{-1} |0\rangle \right\| < \varepsilon \in \mathbb{R}_{>0}$ and outputs a 1 when measuring conditioned on the first qubit. If A is given by an oracle, we may refer to the algorithm as being *relativizing*. We say a classical algorithm solves this problem, if it outputs the vector $|x\rangle$.

This definition follows [3, p. 12]. There are also more general definitions for relativizing algorithms [59, pp. 376-377], but this notion suffices for this text.

Theorem C.6. MATRIXINVERSION is BQP-complete.

The proof of this theorem can be found in [3, p. 4]. Any quantum computation can thus be expressed via an SLE after a polynomial quantum reduction. Harrow et al. then also have the following result [3, pp. 12-14].

Theorem C.7. The following statements hold. Throughout this theorem, the error cap ε of any algorithm here shall be fixed.

(i) If there is a quantum algorithm for solving MATRIXINVERSION with time complexity

$$\kappa^{1-\delta} \, \text{poly}(\log_2(N)) \tag{C.0.3}$$

with $\delta \in (0, 1)$, then BQP = PSPACE.

(ii) No relativizing quantum algorithm for MATRIXINVERSION can run in time $\kappa^{1-\delta} \, \text{poly}(\log_2(N))$.

(iii) If there exists a classical algorithm for MATRIXINVERSION with a runtime of $\text{poly}(\kappa, \log_2(N))$, then BQP = BPP.

(iv) No relativizing classical algorithm for MATRIXINVERSION can run in time $N^\alpha 2^{\beta\kappa}$, unless $3\alpha + 4\beta \geq 1/2$ for any $\alpha, \beta \in \mathbb{R}_{>0}$.

The error cap is fixed to e.g. 1/100, because we consider the other parameters in these results. We want to discuss the first and third statement. For the first part of the theorem, the first direction is already proven via (i) in Theorem C.4. Taking a problem in PSPACE, it can be polynomially reduced to the problem of TQBF and then, using so-called *exhaustive enumeration*, an associated formula can be used to obtain an instance of the MATRIXINVERSION problem. Given the stated complexity, we can then derive the claimed polynomial reduction, giving PSPACE $\subseteq$ BQP. This, however, is an open problem. For the third statement, we again have an open problem in the direction BQP $\subseteq$ BPP. The proof is analogous to the proof of the first statement.

Theorem C.8. The following statements hold.

(i) If there is a quantum algorithm for solving MATRIXINVERSION in time

$$\mathrm{poly}(\kappa, \log_2(N), \log_2(1/\varepsilon)) \tag{C.0.4}$$

then BQP = PP.

(ii) No relativizing algorithm for MATRIXINVERSION can run in time $\mathcal{O}(N^\alpha \,\mathrm{poly}(\kappa)/\varepsilon^\beta)$ for $\alpha, \beta \in \mathbb{R}_{>0}$, unless $\alpha + \beta \geq 1$.

This result can be found in [3, p. 14]. We consider the first statement only. Again, we already have BQP $\subseteq$ PP and want to prove PP $\subseteq$ BQP. The authors use the PP-complete problem #SAT, which counts the fulfilling assignments of the variables in a given Boolean formula φ of $n \in \mathbb{N}_{\geq 1}$ variables and reduce it to a problem instance of MATRIXINVERSION. One major point in the proof is, that the $\log_2(N)$ runtime term mitigates the complexity of thte size of the reduced equation system and the $\log_2(1/\varepsilon)$ factor mitigates the complexity induced by chosing an exponentially small error.

Remark C.9. One may now question the validity of the result by Childs et al. in Section 3.5. However, as Childs et al. point out [40, p. 2], the measurement of the first qubit as in MATRIXINVERSION is a crucial difference in the design of the different algorithms, making a the possibility of a subexponential error algorithm of this form unlikely.

References

[1] D. J. Griffiths and D. F. Schroeter, *Introduction to Quantum Mechanics*. Cambridge University Press, Aug. 2018. DOI: 10.1017/9781316995433.

[2] D. Werner, *Funktionalanalysis*. Springer Berlin Heidelberg, 2018. DOI: 10.1007/978-3-662-55407-4.

[3] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for solving linear systems of equations," *Phys. Rev. Lett. vol. 15, no. 103, pp. 150502 (2009)*, Nov. 19, 2008. DOI: 10.1103/PhysRevLett.103.150502. arXiv: 0811.3171 [quant-ph].

[4] Y.-A. Chen and X.-S. Gao, "Quantum Algorithms for Boolean Equation Solving and Quantum Algebraic Attack on Cryptosystems," Dec. 18, 2017. arXiv: 1712.06239 [quant-ph].

[5] J. Ding, V. Gheorghiu, A. Gilyén, S. Hallgren, and J. Li, "Limitations of the Macaulay matrix approach for using the HHL algorithm to solve multivariate polynomial systems," 2021. DOI: 10.48550/ARXIV.2111.00405.

[6] W. Scherer, *Mathematics of Quantum Computing*. Springer International Publishing, 2019. DOI: 10.1007/978-3-030-12358-1.

[7] M. A. Nielsen, *Quantum computation and quantum information*, 10th ed. Cambridge University Press, 2010, ISBN: 9781107002173.

[8] O. Forster, *Analysis 2, Differentialrechnung im IRn, gewöhnliche Differentialgleichungen*. Springer Spektrum, 2017, ISBN: 9783658194109.

[9] G. Fischer and B. Springborn, *Lineare Algebra*. Springer Berlin Heidelberg, 2020. DOI: 10.1007/978-3-662-61645-1.

[10] K. Janich, *Lineare Algebra*. Springer, 2010, ISBN: 9783540755012.

[11] T. Lyche, *Numerical Linear Algebra and Matrix Factorizations*. Springer, ISBN: 9783030364670.

[12] D. Dervovic, M. Herbster, P. Mountney, S. Severini, N. Usher, and L. Wossnig, "Quantum linear systems algorithms: a primer," Feb. 22, 2018. arXiv: 1802.08227 [quant-ph].

[13] S. Waldmann, *Lineare Algebra 2*. Springer Berlin Heidelberg, 2022. DOI: 10.1007/978-3-662-63639-8.

[14] M.-D. Choi, "Tricks or Treats with the Hilbert Matrix," *The American Mathematical Monthly*, vol. 90, no. 5, May 1983. DOI: 10.2307/2975779.

[15] H. S. Wilf, *Finite Sections of Some Classical Inequalities*. Springer Berlin Heidelberg, 1970. DOI: 10.1007/978-3-642-86712-5.

[16] J. Todd, "The Condition Number of the Finite Segment of the Hilbert Matrix," *Washington, U. S. Govt. Print. Off.*, Contributions to the solution of systems of linear equations and the determination of eigenvalues. O. Taussky, Ed., pp. 109–116, 1954.

[17] G. Fischer, *Lehrbuch der Algebra, Mit lebendigen Beispielen, ausführlichen Erläuterungen und zahlreichen Bildern*. Springer Spektrum, 2017, ISBN: 9783658193652.

[18] S. Arora and B. Barak, *Computational Complexity A Modern Approach - Internet Draft, A Modern Approach*. 2007. [Online]. Available: https://theory.cs.princeton.edu/complexity/book.pdf.

[19] M. Homeister, *Quantum Computing verstehen*. Springer Fachmedien Wiesbaden, 2018. DOI: 10.1007/978-3-658-22884-2.

[20] P. Kaye and M. Mosca, "Quantum Networks for Generating Arbitrary Quantum States," *Phillip Kaye, Michele Mosca, "Quantum Networks for Generating Arbitrary Quantum States", Proceedings, International Conference on Quantum Information (ICQI). Rochester, New York, USA, 2001*, Jul. 14, 2004. arXiv: quant-ph/0407102 [quant-ph].

[21] M. Mottonen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, "Transformation of quantum states using uniformly controlled rotations," *Quant. Inf. Comp. 5, 467 (2005)*, Jul. 1, 2004. arXiv: quant-ph/0407010 [quant-ph].

[22] D. Aharonov and A. Ta-Shma, "Adiabatic Quantum State Generation and Statistical Zero Knowledge," 2003. DOI: https://doi.org/10.48550/arXiv.quant-ph/0301023.

[23] L. Grover and T. Rudolph, "Creating superpositions that correspond to efficiently integrable probability distributions," Aug. 15, 2002. arXiv: quant-ph/0208112 [quant-ph].

[24] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry*. Springer Berlin Heidelberg, 2008. DOI: 10.1007/978-3-540-77974-2.

[25] B. Parhami and M. McKeown, "Arithmetic with binary-encoded balanced ternary numbers," Nov. 2013. DOI: 10.1109/acssc.2013.6810470.

[26] P. Gokhale, J. M. Baker, C. Duckering, N. C. Brown, K. R. Brown, and F. T. Chong, "Asymptotic Improvements to Quantum Circuits via Qutrits," May 24, 2019. DOI: 10.1145/3307650.3322253. arXiv: 1905.10481 [quant-ph].

[27] D. J. Griffiths, *Introduction to Electrodynamics*. Cambridge University Press, ISBN: 9781108420419.

[28] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, "Quantum Amplitude Amplification and Estimation," *Quantum Computation and Quantum Information, Samuel J. Lomonaco, Jr. (editor), AMS Contemporary Mathematics, 305:53-74, 2002*, May 15, 2000. DOI: 10.1090/conm/305/05215. arXiv: quant-ph/0005055 [quant-ph].

[29] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, "Tight Bounds on Quantum Searching," *Fortschritte der Physik*, vol. 46, no. 4-5, pp. 493–505, Jun. 1998. DOI: 10.1002/(sici)1521-3978(199806)46:4/5<493::aid-prop493>3.0.co;2-p.

[30] D. W. Berry, A. M. Childs, and R. Kothari, "Hamiltonian simulation with nearly optimal dependence on all parameters," *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science (FOCS 2015), pp. 792-809 (2015)*, Jan. 8, 2015. DOI: 10.1109/FOCS.2015.54. arXiv: 1501.01715 [quant-ph].

[31] Q. Zhao, Y. Zhou, A. F. Shaw, T. Li, and A. M. Childs, "Hamiltonian simulation with random inputs," Nov. 8, 2021. arXiv: 2111.04773 [quant-ph].

[32] Y. Cao *et al.*, "Quantum Chemistry in the Age of Quantum Computing," *Chemical Reviews*, vol. 119, no. 19, pp. 10 856–10 915, Aug. 2019. DOI: 10.1021/acs.chemrev.8b00803.

[33] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, "Efficient quantum algorithms for simulating sparse Hamiltonians," *Communications in Mathematical Physics 270, 359 (2007)*, Aug. 18, 2005. DOI: 10.1007/s00220-006-0150-x. arXiv: quant-ph/0508139 [quant-ph].

[34] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, 2009, ISBN: 9780262033848.

[35] M. Suzuki, "Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations," *Physics Letters A*, vol. 146, no. 6, pp. 319–323, Jun. 1990. DOI: 10.1016/0375-9601(90)90962-n.

[36] S. J. Richard, *An Introduction to the conjugate gradient method without the agonizing pain*, 1994. DOI: 10.1.1.110.418.

[37] K. Königsberger, *Analysis 1*. Springer, 2003, ISBN: 978-3-540-40371-5.

[38] O. Forster, *Analysis 1*. Springer Fachmedien Wiesbaden, 2016. DOI: 10.1007/978-3-658-11545-6.

[39] Aigner, *Diskrete Mathematik*. Vieweg, 2006. DOI: 10.1007/978-3-8348-9039-9.

[40] A. M. Childs, R. Kothari, and R. D. Somma, "Quantum algorithm for systems of linear equations with exponentially improved dependence on precision," *SIAM Journal on Computing 46, 1920-1950 (2017)*, Nov. 7, 2015. DOI: 10.1137/16M1087072. arXiv: 1511.02306 [quant-ph].

[41] V. Kasirajan, *Fundamentals of Quantum Computing*. Springer International Publishing, 2021. DOI: 10.1007/978-3-030-63689-0.

[42] Y. Lee, J. Joo, and S. Lee, "Hybrid quantum linear equation algorithm and its experimental test on IBM Quantum Experience," *Scientific Reports*, vol. 9, no. 1, Mar. 2019. DOI: 10.1038/s41598-019-41324-9. arXiv: 1807.10651 [quant-ph].

[43] Y. Cao, A. Daskin, S. Frankel, and S. Kais, "Quantum Circuit Design for Solving Linear Systems of Equations," Oct. 10, 2011. DOI: 10.1080/00268976.2012.668289. arXiv: 1110.2232 [quant-ph].

[44] A. M. Childs and R. Kothari, "Simulating sparse Hamiltonians with star decompositions," *Theory of Quantum Computation, Communication, and Cryptography (TQC 2010), Lecture Notes in Computer Science 6519, pp. 94-103 (2011)*, pp. 94–103, Mar. 18, 2010. DOI: 10.1007/978-3-642-18073-6_8. arXiv: 1003.3683 [quant-ph].

[45] S. K. Leyton and T. J. Osborne, "A quantum algorithm to solve nonlinear differential equations," Dec. 23, 2008. arXiv: 0812.4423 [quant-ph].

[46] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles, "Variational Quantum Linear Solver," Sep. 12, 2019. arXiv: 1909.05820 [quant-ph].

[47] A. Ambainis, "Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations," 2010. DOI: 10.48550/ARXIV.1010.4458.

[48] M. Dworkin *et al.*, "Advanced Encryption Standard (AES)," *Federal Inf. Process. Stds. (NIST FIPS), National Institute of Standards and Technology, Gaithersburg, MD*, Nov. 26, 2001. DOI: https://doi.org/10.6028/NIST.FIPS.197.

[49] J. Daemen and V. Rijmen, *The Design of Rijndael*. Springer Berlin Heidelberg, 2020. DOI: 10.1007/978-3-662-60769-5.

[50] H. Nover, "Algebraic Cryptanalysis of AES: An Overview," [Last accessed: 04.10.2022, 14:24]. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.81.5435.

[51] S. Murphy and M. J. B. Robshaw, "Essential Algebraic Structure within the AES," in *Advances in Cryptology — CRYPTO 2002*, Springer, Jan. 1, 2002, ISBN: 978-3-540-44050-5. DOI: 10.1007/3-540-45708-9_1.

[52] C. Karpfinger and K. Meyberg, *Algebra*. Springer Berlin Heidelberg, 2017. DOI: 10.1007/978-3-662-54722-9.

[53] A. Kaminsky, M. Kurdziel, and S. Radziszowski, "An overview of cryptanalysis research for the advanced encryption standard," in *2010 - MILCOM 2010 MILITARY COMMUNICATIONS CONFERENCE*, IEEE, Oct. 2010. DOI: 10.1109/milcom.2010.5680130.

[54] J. Gao, H. Li, B. Wang, and X. Li, "Quantum security of AES-128 under HHL algorithm," *Quantum Information and Computation*, vol. 22, pp. 209–240, Feb. 2022. DOI: 10.26421/QIC22.3-4-2.

[55] N. Courtois and J. Pieprzyk, "Cryptanalysis of Block Ciphers with Overdefined Systems of Equations," *Cryptology ePrint Archive, Report 2002/044*, 2002, https://ia.cr/2002/044. DOI: https://doi.org/10.1007/3-540-36178-2_17.

[56] R.-H. Schulz, *Codierungstheorie eine Einführung, eine Einführung*. Vieweg, 2003, ISBN: 978-3528164195.

[57] K. Jänich, *Funktionentheorie, Eine Einführung (Springer-Lehrbuch)*. Springer, 2004, ISBN: 9783540203926.

[58] R. Remmert and G. Schumacher, *Funktionentheorie 1*. Springer Berlin Heidelberg, 2002. DOI: 10.1007/978-3-642-56281-5.

[59] M. Sipser, *Introduction to the theory of computation*. Cengage Learning, 2013, ISBN: 113318779X.

[60] L. M. Adleman, J. DeMarrais, and M.-D. A. Huang, "Quantum computability," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1524–1540, Oct. 1997. DOI: 10.1137/S0097539795293639.