



# Quantencomputing in der industriellen Applikation:

**Vom Algorithmen-, Markt- und  
Hardwareüberblick hin zu  
anwendungszentriertem  
Quantensoftware-Engineering**

---

Fraunhofer IAO, Fraunhofer IPA,  
Fraunhofer IAF,  
FZI Forschungszentrum Informatik,  
Universität Stuttgart IAAS,  
Eberhard Karls Universität Tübingen,  
Universität Stuttgart HLRS

# IMPRESSUM

## Kontaktadresse

Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO  
Nobelstraße 12, 70569 Stuttgart  
Projektwebsite: <https://websites.fraunhofer.de/sequoia/>

Dr. Christian Tutschku  
Telefon: +49 152 28835266  
[christian.tutschku@iao.fraunhofer.de](mailto:christian.tutschku@iao.fraunhofer.de)

## Autoren

Dr. Christian Tutschku	Fraunhofer IAO
Dr. Andreas Sturm	Fraunhofer IAO
Dr. Florian Knäble	Fraunhofer IAO
Dr. Bharadwaj Mummaneni	Fraunhofer IAO
Daniel Pranjić	Fraunhofer IAO
Chiara Stephan	Fraunhofer IAO
Daniel Mayer	Fraunhofer IAO
Gereon Koßmann	Fraunhofer IAO
Dr. Marco Roth	Fraunhofer IPA
Dr. Paul-Amaury Matt	Fraunhofer IPA
Arthur Grigorjan	Fraunhofer IPA
Dr. Thomas Wellens	Fraunhofer IAF
Kathrin König	Fraunhofer IAF
Oliver Denninger	FZI Forschungszentrum Informatik
Martin Beisel	Universität Stuttgart IAAS
Felix Truger	Universität Stuttgart IAAS
Simon Garhofer	Eberhard Karls Universität Tübingen
Dr. Farida Shagieva	Universität Stuttgart HLRS

## Titelbild

© Carolin Lehmann

## Fraunhofer-Publica

[10.24406/publica-805](https://www.fraunhofer.de/publica/10.24406/publica-805)

## Gefördert durch



Baden-Württemberg

MINISTERIUM FÜR WIRTSCHAFT, ARBEIT UND TOURISMUS



CC-BY-NC-ND-Lizenz

## MANAGEMENT SUMMARY

In jüngster Zeit erfährt die Schlüsseltechnologie Quantencomputing breites Interesse in Forschung und Wirtschaft. Dies basiert auf ihrem großen industriellen Anwendungspotenzial und den perspektivisch zahlreichen Einsatzmöglichkeiten. Bevor Unternehmen Quantenapplikationen jedoch nutzenbringend einsetzen können, bedarf es einer Entwicklung von praxistauglichen und anwendungsspezifischen Quantenalgorithmen. Diese Studie gibt einen umfassenden Überblick über die Bereiche Quantenhardware und -software und präsentiert Ergebnisse aus den Anwendungsfällen im Verbundforschungsprojekt »SEQUOIA – Software-Engineering industrieller, hybrider Quantenanwendungen und -algorithmen«. Die Studie vermittelt Akteuren im universitären und angewandten Technologiebereich ein Verständnis für die wirtschaftliche Nutzung von Quantencomputing, sowohl auf technischem Level, als auch managementbezogen.

**Sechs Anwendungsfälle** zeigen mögliche quantengestützte Herangehensweisen auf und diskutieren die Qualität heutiger NISQ-Umsetzungen (Noisy-Intermediate-Scale-Quantum) sowie notwendige Schritte hin zu **performanten End-to-End Lösungen**. Die diskutierten industriellen Anwendungsbeispiele zeigen, dass mittels Quantenalgorithmen auf dem IBM Quantum System One in Ehningen (Stuttgart) durch **gezielte Fehlermitigationstechniken und Optimierungsalgorithmen** bereits heute **erste Proof-of-Concept-Resultate** erzielt werden können. Neben der Qubitanzahl spielen der Einsatz von klassischen Preprocessing-Methoden und die Wahl der Parameter wie die Tiefe der Schaltkreise, die Größe der linearen Systeme und der Umfang der Problem instanzen eine wesentliche Rolle, um sinnvolle Lösungen zu erhalten. Trotz heutiger Limitationen besitzt die Anwendung von (hybriden) Quantenalgorithmen vielversprechendes Potenzial, um neue Lösungsansätze für rechenintensive Probleme zu entwickeln.

Mit dem **SEQUOIA-Anwendungsbaukasten** werden Entwickelnde bei der Konstruktion und Ausführung von Quantenlösungen in der Praxis unterstützt. Hierfür werden zentrale Werkzeuge, Programmierschnittstellen und Komponenten für das Quantensoftware-Engineering vorgestellt, um Quantenalgorithmen gezielt und nutzenbringend einzusetzen. Die entwickelten Anwendungslösungen werden in die nationale Plattform des Verbundprojekts PlanQK integriert und sind als **interaktive Jupyter Notebooks** auf der SEQUOIA-Projektwebsite verfügbar.

# INHALT

<b>1</b>	<b>Das Quantencomputing Ökosystem</b>	<b>1</b>
1.1	Einleitung	1
1.2	Grundlagen	8
1.2.1	Das Bit und das Qubit	8
1.2.2	Darstellung auf der Bloch-Sphäre	9
1.2.3	Qubits messen	11
1.2.4	Multi-Qubit-Systeme	15
1.3	Hardwarerealisierungen von Quantencomputern	18
1.3.1	Ionenfallen Technologie	21
1.3.2	Supraleitende Architekturen	23
1.3.3	Stickstoffvakanz-Zentren in Diamant	28
1.3.4	Photonen	34
1.4	Quantenalgorithmen	39
1.4.1	Frühe Quantenalgorithmen	39
1.4.2	Quanten-Fouriertransformation	39
1.4.3	Algorithmen für fehlerkorrigierte Quantencomputer	41
1.4.4	Variationelle Quantenalgorithmen	44
1.5	Exkurs: Quantenfehlerkorrektur	48
<b>2</b>	<b>Die SEQUOIA Anwendungsfälle</b>	<b>53</b>
2.1	Das assoziierte Unternehmensnetzwerk	53
2.2	Anwendungsfälle	54
2.2.1	Fokusanwendung 1 »Optimierung von Ladeplänen einer gegebenen Ladesäuleninfrastruktur«	54
2.2.2	Fokusanwendung 2 »Quantenbasierte numerische Strömungssimulation«	60
2.2.3	Fokusanwendung 3 »Optimierung einer Auftragssequenzierung in der diskreten Fertigung«	65

2.2.4	Fokusanwendung 4 »Optimale Auslegung von Schnittmustern in der Blechfertigung«	69
2.2.5	Fokusanwendung 5 »Szenario-basierte Routenplanung zur Absicherung von Fahrfunktionen«	74
2.2.6	Fokusanwendung 6 »Routenplanung von LKW-Flotten im Supply-Chain-Management«	77
2.3	<b>Expectation Management und Handlungsempfehlungen</b>	<b>81</b>
<b>3</b>	<b>Der SEQUOIA Anwendungsbaukasten – Quantencomputing in der Praxis</b>	<b>83</b>
3.1	<b>Quantensoftware-Engineering</b>	<b>84</b>
3.1.1	Vorgehensmodelle	85
3.1.2	Quantumsoftware-Lifecycle	86
3.1.3	Bedeutung von Simulatoren	91
3.1.4	Cloud-Backends	91
3.2	<b>Der SEQUOIA Werkzeugkasten - High-Level Entwickler API</b>	<b>93</b>
3.2.1	Schaltkreisgenerierungs-Modul	93
3.2.2	Übersetzungs-Modul	94
3.2.3	Ausführungs-Modul	94
3.2.4	Fehlermitigations-Modul	95
3.2.5	Automatisierung von Quantenanwendungen	95
3.2.6	Erweiterbarkeit	96
3.2.7	Relation zum NISQ Analyzer	96
3.3	<b>Beispiel anhand eines Use Cases</b>	<b>97</b>
<b>4</b>	<b>Wissenstransfer und Ökosystem</b>	<b>100</b>
	<b>Literaturverzeichnis</b>	<b>103</b>
	<b>Abbildungsverzeichnis</b>	<b>114</b>
	<b>Tabellenverzeichnis</b>	<b>118</b>
	<b>Infoboxenverzeichnis</b>	<b>119</b>



# 1 DAS QUANTENCOMPUTING ÖKOSYSTEM

## 1.1 Einleitung

Quantentechnologie im Allgemeinen sowie Quantencomputing (QC) im Speziellen hat in den letzten Jahren massives Interesse auf sich gezogen. Als eine der vielversprechendsten Innovationen in naher Zukunft erhält jener neuartige Ansatz des Computings bereits heute Einzug in Wissenschaft und Wirtschaft. Dies geschieht mit dem langfristigen Ziel performante Lösungen für komplexeste, bisher gar unlösbare Probleme zur Verfügung zu stellen. Im Folgenden werden sowohl die Eckpunkte derzeitiger Aktivitäten im deutschen Quantencomputing Ökosystem diskutiert als auch ein Überblick über wichtige aktuelle<sup>1</sup> Quantencomputing Hard- und Softwareentwicklungen bereitgestellt. Jenes einleitende Kapitel schließt mit einer Kurzbeschreibung des hiesigen Studienaufbaus.

### **Die NISQ-Ära**

Bereits heute sind erste Quantencomputer in der Lage einfache Problemstellungen zu lösen. Zwar ist langfristig das Quantencomputingpotenzial für industrielle Anwendungen seit einigen Jahren bekannt, zeitgemäße Softwarelösungen müssen allerdings die noch hohe Fehleranfälligkeit heutiger QC-Hardwareansätze berücksichtigen, welche die sogenannte Noisy-Intermediate-Scale-Quantum-Ära (NISQ) definiert. Teils geschieht dies durch die Konzipierung von hybriden (quanten-klassischen) Lösungen. Auf dem Weg hin zu fehlerkorrigierten Post-NISQ-Quantensystemen, kennzeichnet den aktuellen Stand der Technik vor allem, dass Quantenalgorithmen mit klassischen Algorithmen kombiniert werden. Dies geschieht, um den heutigen Quanteninformationsverlust mittels hybrider Lösungen zu umgehen und bereits kurzfristig die Potenziale von QC auszuschöpfen. In jenen Ansätzen werden also iterativ klassische Komponenten in Quantenalgorithmen eingebunden, um somit nur (die schwierigsten) Teile einer komplexen Problemstellung von Quantencomputern berechnen zu lassen. Je leistungsfähiger die Quantenhardware, desto längere und mächtigere Quantenalgorithmen können auf dem Weg in die Post-NISQ-Quantenära umgesetzt werden.

### **Hardware- und Softwareentwicklung**

In Deutschland verzeichnen wir bundesweit bereits erste Meilensteine in der Hardwareentwicklung. Mit dem »IBM Quantum System One« wurde in Ehningen bei Stuttgart 2021 der erste kommerziell nutzbare, DSGVO-konforme sowie gatterbasierte Quantencomputer in Europa offiziell in Betrieb genommen. Dieser arbeitet aktuell (Stand

---

<sup>1</sup> Dezember 2022

### Infobox 1: **Quantenvolumen und CLOPS**

Das Quantenvolumen (QV) ist eine Metrik, die die Fähigkeit eines Quantencomputers charakterisiert Quantenschaltkreise mit hoher Güte auszuführen. Konkret definiert das QV die maximale Größe zufällig erzeugter quadratischer Quantenschaltkreise (Anzahl der Gate-Layer ist gleich Anzahl der Qubits), bis zu welcher diese aufgrund von Fehlerraten erfolgreich, d.h. mit einer hohen Güte, ausgeführt werden können. Die Form der Schaltkreise ist dabei unabhängig von der Architektur des Quantencomputers. Somit ist das QV eine Metrik, mit der unterschiedliche Quantencomputerarchitekturen miteinander verglichen werden können

[1].

Die CLOPS (Circuit Layer Operations per Second) ist eine Metrik, die die Rechengeschwindigkeit eines Quantencomputers definiert. Klassisches Gegenstück dazu ist die sogenannte FLOPS (Floating Point Operations per Second). Konkret ist die CLOPS definiert als die Anzahl der QV-Schaltkreis-Layer, die pro Sekunde ausgeführt werden können. Die Ausführungszeit inkludiert dabei neben der reinen Rechenzeit der QPU auch die zugehörigen Zeiten zur klassischen Kommunikation sowie Vorbereitung

Dezember 2022) mit einem 27-Qubit Falcon r5.11 Prozessor und besitzt ein Quantenvolumen von 64 (bzw. eine CLOPS von 1.9K) (Infobox 1).

Auf dem [IBM Quantum Summit 2022](#) wurde unterdessen in New York City der 433 Qubit Osprey Chip vorgestellt, welcher Forschenden Anfang 2023 für erste Rechnungen zur Verfügung stehen soll. Gleichzeitig wurde die sogenannte Methode *»Dynamic Circuits«* verfügbar gemacht, welche bis dato nur simulierbar war und einen Meilenstein hin zu performanten industriellen Anwendungen von QC definiert. Insbesondere wird jene Methoden benötigt um *Quantum Convolutional Neural Networks* (QCNNs) oder Methoden der Quantenfehlerkorrektur zu ermöglichen, welche das Fundament für Post-NISQ-Algorithmen darstellt. Des Weiteren wurden wichtige algorithmische sowie anwendungsbezogene Softwaremodule in Form von Quantum Services abgeschlossen, die durch intelligente Schaltkreisbibliotheken Quantenanwendungen, insbesondere für maschinelles Lernen und Optimierungsprobleme, in ihrer Komplexität reduzieren sollen. Abschließend wurde die [IBM Roadmap](#) hin zu gekoppelten Quantum Processing Units (QPUs) auf dem Weg zum eingeführten *»IBM Quantum System Two«* definiert. In den nächsten Jahren soll die Qubitzahl und deren Qualität weiter vervielfacht werden. Nach 2026 hat es sich IBM zum Ziel gesetzt, ihre Prozessoren mit 10-100.000 Qubits auszustatten, was bei reduzierten Fehlerraten Konzepte der Fehlerkorrektur ermöglichen würde.



Des Weiteren wurde am Forschungszentrum Jülich jüngst der erste sogenannte Quantenannealer mit mehr als 5000 supraleitenden Qubits in Betrieb genommen und in das bestehende HPC-Cluster integriert. Bei jenem Annealer der kanadischen Firma D-Wave handelt es sich nicht um einen universellen Quantencomputer ähnlich der IBM Systeme. Ein direkter Vergleich der Qubitzahlen sollte also nicht ohne weiteres Hintergrundwissen gezogen werden. Jener Hardwareansatz wurde speziell für bestimmte (industrielle) Anwendungsfälle (z.B. für kombinatorische Optimierung) entwickelt und liefert hier bereits [1 3],[1 4], [1 5] beeindruckende Ergebnisse.

Neben Integration der obigen Systeme aus Kanada sowie den USA in das deutsche Ökosystem ist es allerdings erklärtes Ziel der Bundesregierung, selbst mittel- und vor allem langfristig einen international wettbewerbsfähigen Quantenrechner mit mindestens 100 Qubits zu entwickeln. Dies wurde 2020 in der zugrundeliegenden »Roadmap Quantencomputing« definiert. Damit soll ein sogenannter Quantenvorteil (englisch Quantum Advantage) für praxisrelevante Anwendungen demonstriert werden. Um Deutschland zum Vorreiter in diesem Bereich zu machen, unterstützte die Bundesregierung jenes Vorhaben zur Erforschung von Quantentechnologien mit 650 Millionen Euro und weiteren zwei Milliarden Euro aus dem Konjunktur- und Zukunftspaket im Juni 2020.

Mit selbigem Ziel verabschiedete das DLR im Rahmen seiner Quantencomputing-Initiative im Oktober 2022 Aufträge im Gesamtvolumen von 208.5 Millionen Euro zur Realisierung von Qubits basierend auf Ionenfallen. Bis zum Ende der Projekte sollen die geförderten Unternehmen eleQtron, NXP® Semiconductors Germany, Parity Quantum Computing Germany, QUDORA Technologies und Universal Quantum Deutschland Ionenfallen-Quantencomputer mit mindestens 50 Qubits installieren, die einen ersten Schritt hin zu fehlertolerantem Computing ermöglichen. Weiter sollen modulare Systeme aufgebaut werden, die eine Erweiterung in Richtung 1000 Qubits ermöglichen.

Es ist ebenfalls erklärtes Ziel des Munich Quantum Valleys im Rahmen des Zentrums für Quantencomputing und -technologien (ZQQ) bereits kurzfristig (ca. 5 Jahre) eine NISQ Hardware mit 50-100 Qubits zu entwickeln und langfristig (ca. 10 Jahre) skalierbare und kommerziell nutzbare Quantencomputer, basierend auf Atomen/Ionen und supraleitenden Qubits zu bauen und in Betrieb zu nehmen. Dazu investierte der Freistaat Bayern rund 70 Millionen Euro im Rahmen der Hightech Agenda Bayern, sowie plante weiter im Beschleunigungsprogramm Hightech Agenda Plus 2021 und 2022 je bis zu 60 Millionen Euro für die Förderung der Quantentechnologien im Staatshaushalt ein.

## Projektaktivitäten im deutschen Ökosystem

Diese Beispiele unterstreichen eindrucksvoll die Relevanz von Quantencomputing für die deutsche Wirtschaft, Wissenschaft und Politik, was sich insbesondere in den damit verknüpften angewandten Forschungsprojekten zeigt. Das Projekt PhotonQ soll hier exemplarisch aufgeführt werden. Hierbei handelt es sich um ein Verbundprojekt für photonische Quantenprozessoren unter der Leitung der Universität Stuttgart, das mit 16 Millionen Euro vom Bundesministerium für Bildung und Forschung (BMBF) gefördert wurde. Auch das Verbundprojekt QC4DB der Universität zu Lübeck mit dem Verbundpartner Quantum-Brilliance – ein deutsch-australisches Start-up sowie einer der führenden Hersteller von NV-basierter Quantenhardware – erfährt mit fast zwei Millionen Euro eine zielgerichtete finanzielle Förderung durch das BMBF. Darüber hinaus gibt es derzeit eine Vielzahl an weiteren Forschungsprojekten (ProvideQ, EniQma, AutoQML, Aquas, QuiX, ...), Start-up-Gründungen (HQS Quantum Simulations, JoS Quantum, Terra Quantum, Kiutra, Quanticor Security, Quantum Factory, Kipu Quantum, NVision, Q.ANT, QMware, Quantum Optics Jena, Quartiq, KEEQuant, Avanetix, uvm.) im Quantenbereich oder Initiativen der Wirtschaft selbst (QUTAC, Quantum Business Network QBN, etc.).

Insbesondere treibt die Fraunhofer-Gesellschaft mit dem nationalen Kompetenznetzwerk Quantencomputing, bestehend aus sieben regionalen Zusammenschlüssen (sogenannten Kompetenzzentren), die Entwicklung sowohl auf Hardware- als auch auf Softwareseite aktiv voran.

In Baden-Württemberg wurde 2020 das »Kompetenzzentrum Quantencomputing BW« (KQCBW) durch die Landesregierung ins Leben gerufen. Gefördert vom Ministerium für Wirtschaft, Arbeit und Tourismus unterstützt dieses Förderprogramm die zielgerichtete Forschungs- und Entwicklungsarbeit zur praktischen Anwendung von Quantencomputern und wird durch das Fraunhofer IAF in Freiburg sowie durch das Fraunhofer IAO in Stuttgart geleitet und koordiniert.

Dabei stehen die folgenden drei Schwerpunkte im Zentrum:

- Entwicklung und Demonstration leistungsfähiger Quantenspeicher und -prozessoren
- Quantencomputing für das Design neuartiger Materialien und quantenchemischer Reaktionen
- Quantencomputer und -algorithmen zur Optimierung komplexer Zustandssysteme

In der ersten Projektphase des KQCBW wurden mit obiger Zielsetzung insgesamt sechs Verbundprojekte gefördert, welche von verschiedenen Fraunhofer-Instituten und (außer-) universitären Forschungseinrichtungen gemeinsam bearbeitet werden und

unterschiedlichen Software- bzw. Hardwarefokus besitzen: SEQUOIA, EFFEKTIF, QORA, QC-4-BW, SiQuRE und QuEst. Das Projekt SEQUOIA definiert dabei das größte Softwareverbundvorhaben. Dessen Erkenntnisse und Projektergebnisse stellen maßgeblich den Inhalt der vorliegenden Studie dar.

Neben jenen genannten Software- und Hardwareprojekten entsteht seit 2020 am Standort des »Quantum System One« auf dem IBM Campus bei Stuttgart ebenfalls das sogenannte »Quantum Village Ehningen«. Im (Pilot)Projekt Q.AX, welches im Oktober 2022 begonnen hat, wird hier ein lokales Zentrum für Quantentechnologien und Künstliche Intelligenz aufgebaut, welches branchenübergreifend Unternehmen und Fachkundige räumlich zusammenbringt und dem (Quanten)Innovationsstandort Baden-Württemberg damit eine Strahlkraft weit über die nationalen Grenzen hinaus verleiht. Langfristig sollen an jenem Standort die »Quantum Gardens« entstehen, ein Bauvorhaben, das die Nutzung des (ehemaligen) IBM-Areals als gemischten Wohn-Tech-Campus vorsieht und Ehningen damit als zentrale Anlaufstelle für Quantencomputing im Land Baden-Württemberg etabliert. Diesem Vorhaben liegt eine Studie des Fraunhofer IAO zugrunde, welche die Potenziale und Chancen eines derartigen urbanen Quartiers am Standort Ehningen erforscht.

### **Anwendungszentriertes Quantensoftware-Engineering:**

#### **Die Projekte SEQUOIA und PlanQK**

Um das Potenzial von Quantencomputern anwendungszentriert zu erschließen, sind neben der Hardwareentwicklung insbesondere Methoden und Prozesse der Quantensoftwareentwicklung erforderlich. Innerhalb des KQCBW startete daher Anfang 2021 das Verbundprojekt SEQUOIA, dessen Fokus das Software Engineering industrieller hybrider Quantenanwendungen und -algorithmen ist. Im Projekt werden neue Methoden, Werkzeuge und Ansätze im Bereich des Quantencomputings untersucht, entwickelt sowie getestet, um industrielle Quantenanwendungen zu verbessern und zu beschleunigen. Insgesamt bündelt SEQUOIA als gemeinsames Forschungsprojekt die fachlichen Kompetenzen des Fraunhofer IAO, Fraunhofer IAF sowie Fraunhofer IPA, des FZI Forschungszentrums Informatik, des Lehrstuhls für eingebettete Systeme der Eberhard Karls Universität Tübingen sowie der Universität Stuttgart mit dem Höchstleistungsrechenzentrum HLRS und dem Institut für Architektur von Anwendungssystemen IAAS.

Hauptziel des Verbundforschungsprojekts ist es, in enger Zusammenarbeit mit den Partnerunternehmen hybride Quantenlösungen für industrielle Anwendungsfälle zu entwickeln. Jene Lösungen werden kooperativ konzipiert, auf lokalen oder cloudbasierten QC-Simulatoren getestet und schließlich auf realer Quantenhardware umgesetzt. Diesen Ansatz verfolgend, entwickelt das Projekt SEQUOIA effiziente Quantenlösungen in den

Fokusbereichen Produktion, Maschinenbau, Logistik und Energiewirtschaft. Das im Projekt aufgebaute Unternehmensnetzwerk umfasst heute bereits mehr als 31 Partnerunternehmen, die jenes Vorhaben assoziiert begleiten und maßgeblich zum Erreichen der obigen Ziele beitragen.

Konkret liegt der Projektschwerpunkt auf sechs repräsentativen Anwendungsfällen für die Industrie in Baden-Württemberg, welche zentraler Bestandteil der vorliegenden Studie sind. Dazu gehören kombinatorische Optimierungsprobleme im Kontext von Elektromobilität und nachhaltigen Produktionszyklen, die Lösung von Varianten des TSP-Problems (Traveling Sales Person) im Bereich des Supply-Chain Managements mit dem Ziel einer effizienten Routenoptimierung, sowie die Lösung von (gekoppelten) Differentialgleichungen zur Unterstützung von CFD-Berechnungen (Computational Fluid Dynamics) im Simulationsbereich. Aus softwaretechnischer Sicht ist die Zuverlässigkeit der erarbeiteten Lösungen entscheidend. Zu diesem Zweck befasst sich das Projekt ebenfalls mit der zentralen Frage, wie die Korrektheit und die Qualität der entwickelten Quantenanwendungen sichergestellt werden.

Durch den Fachkräftemangel im IT-Bereich im Allgemeinen sowie im QC-Bereich im Speziellen spielt dabei insbesondere der Automatisierungsgedanke in der Quantensoftwareentwicklung einen herausragenden Gedanken (z.B. in der Schaltkreisgenerierung). Neben den eigens entwickelten Lösungen wurde sich hier mit führenden internationalen Schlüsselakteuren zusammengeschlossen, um internationale Strahlkraft sicherzustellen. So ist beispielsweise das QC-Start-up »Classiq« im letzten Jahr als assoziierter Partner zum SEQUOIA Projekt hinzugestoßen. Die von ihnen entwickelte innovative Softwarelösung ermöglicht eine beschleunigte automatische Übersetzung von abstrakten Quantenalgorithmen in passenden Quantenschaltungen – ein großer Schritt in der Quantensoftwareentwicklung.

Die SEQUOIA Projektergebnisse leisten mit ihrem Fokus auf anwendungsspezifische industrielle Fallbeispiele exemplarisch einen Beitrag zum Verbundprojekt PlanQK. Jenes Projekt dient als Plattform sowie Ökosystem für quantenunterstützte Künstlicher Intelligenz. Im Fokus steht die bedarfsgerechte cloubasierte Nutzung von Quantencomputern für Quantenapplikationen aus verschiedensten Anwendungsgebieten. PlanQK schließt damit die Wertschöpfungskette und vereint alle relevanten Schlüsselakteure: Das sind einerseits die Forschenden und andererseits die potenziell anwendenden Unternehmen mit ihren spezifischen Anwendungsfällen. Die aus dem Projekt SEQUOIA resultierenden Anwendungsfälle der beteiligten Partnerunternehmen leisten hierfür einen bedeutenden Beitrag.

Erklärtes Ziel des PlanQK Projekts ist es, Entwickelnden in Wirtschaft und Wissenschaft niederschwellig die Nutzung von Quantencomputing als zugängliche Dienstleistung zu

ermöglichen. Dazu werden die integrierten Quantenlösungen in standardisierte Anwendungsschnittstellen überführt (sogenannte APIs, englisch = application programming interfaces), um den Zugang zu jener Technologie niederschwellig bereitzustellen und damit Quantencomputing als Service zu etablieren. Ein derartiger anwendungsgetriebener Service ist beispielsweise die Quantenroutenoptimierung für Lieferdienste, die direkt in bestehende Versand-Software integriert werden kann. Dem Outreach und Open-Innovation Gedanke folgend, ermöglicht PlanQK als Plattform außerdem den direkten Austausch zwischen den Anwendern und jeweiligen (Quanten)Expert\*innen, um alle Beteiligten möglichst direkt von der gemeinsamen Expertise profitieren zu lassen.

### **Aufbau der Studie**

Im ersten Kapitel werden die Grundlagen und die Funktionsweise von Quantencomputing, der aktuelle Stand der Technik hinsichtlich Soft- und Hardware sowie die gängigsten Quantenalgorithmen diskutiert und erläutert.

Im zweiten Kapitel folgt eine Beschreibung der zentralen Akteure und relevanten Branchen im Projekt. Dies führt an die Beschreibung der zentralen SEQUOIA Anwendungsfälle heran, welche anschließend ausführlich beschrieben werden. Sowohl die Herangehensweisen, die heutigen NISQ-Lösungen, als auch die nötigen Schritte hin zu performanten End-to-End Lösungen werden detailliert diskutiert. Insbesondere werden jene Ausführungen mit studienbegleitenden interaktiven Jupyter Notebooks gestützt, um Entwickelnden eine Hands-on Erfahrung zu vermitteln.

Das dritte Kapitel adressiert und beantwortet heutige zentrale Fragen des ganzheitlichen Quantensoftware-Engineerings. Werkzeuge und Programmierschnittstellen werden vorgestellt und deren Relevanz für Entwickelnde erläutert.

Schließlich diskutiert das vierte Kapitel aktuelle Möglichkeiten für Unternehmen, sich auf den Einsatz von Quantencomputern vorzubereiten (englisch Quantum Ready), indem frühzeitig das Bewusstsein für jene Technologie (englisch Quantum Awareness) geschaffen wird: »From Quantum Awareness to Quantum Readiness«. Im Rahmen dessen wurden hierfür eine Vielzahl an Weiterbildungsangeboten wie beispielsweise das Fraunhofer IAO/IAF Quantencomputing Schulungsprogramm und weitere Formate entwickelt.

Autoren: Dr. Christian Tutschku (Fraunhofer IAO), Chiara Stephan (Fraunhofer IAO)

Kontakt: [christian.tutschku@iao.fraunhofer.de](mailto:christian.tutschku@iao.fraunhofer.de), [chiara.stephan@iao.fraunhofer.de](mailto:chiara.stephan@iao.fraunhofer.de)

## 1.2 Grundlagen

### 1.2.1 Das Bit und das Qubit

Im klassischen, digitalen Computer ist die grundlegende Informationseinheit das **Bit** (kurz für binary digit). Dieses zeichnet sich dadurch aus, dass es sich immer in genau einem von zwei möglichen Zuständen befindet. Für gewöhnlich werden diese Zustände mit 0 und 1 bezeichnet. Technisch wird ein Bit (zumeist) über einen Transistor realisiert. Dieser ist ein elektronisches Bauelement, welches man sich vereinfacht als Ein/Aus-Schalter vorstellen kann und über welchen man die zwei Zustände »Strom fließt nicht« (Zustand 0) und »Strom fließt« (Zustand 1) erzeugen kann. Heutige Computer bestehen aus Milliarden von Transistoren und jede Information – seien es Zahlen, Textdokumente, E-Mails, Bilder, Videos, usw. – werden als Folgen von Bits gespeichert und verarbeitet.

Analog zum Bit im klassischen Computer haben wir im Quantencomputer das **Qubit** (kurz für Quantenbit). In diesem Abschnitt werden wir es formal beschreiben und in Abschnitt 1.3. seine technische Umsetzung diskutieren. Genau wie das klassische Bit, kann ein Qubit die Zustände  $|0\rangle$  und  $|1\rangle$  annehmen. Wir verwenden hier die in der Quantenmechanik übliche Dirac'sche Notation, in welcher Quantenzustände mit  $|\cdot\rangle$  bezeichnet werden. Im Gegensatz zu einem Bit kann ein Qubit jedoch auch andere Werte als  $|0\rangle$  und  $|1\rangle$  annehmen. Genauer gesagt, kann es sich in einer beliebigen Superposition (d.h. einer Überlagerung) dieser beiden Zustände befinden. Bezeichnen wir den Zustand des Qubits mit  $|\psi\rangle$  können wir eine solche Superposition präzise ausdrücken als

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

wobei  $\alpha$  und  $\beta$  die sogenannten **Amplituden** zu den **Basiszuständen**  $|0\rangle$  bzw.  $|1\rangle$  sind.

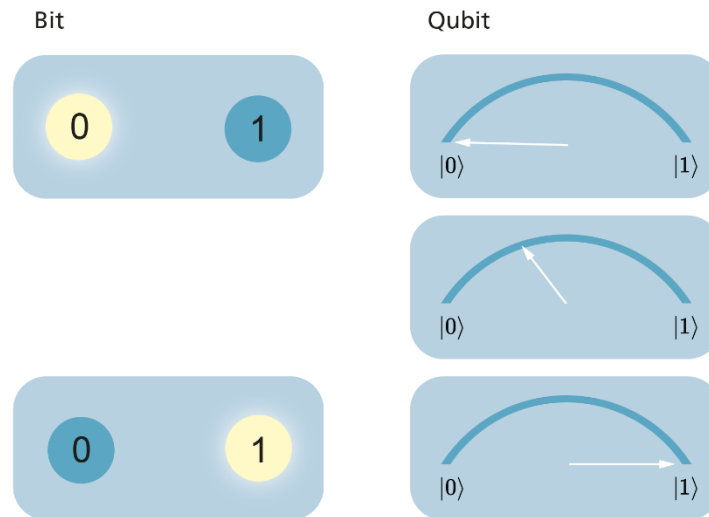


Abbildung 1: Vergleich der möglichen Zustände eines Bits und eines Qubits. Die Darstellung beim Qubit ist aus Gründen der Niederschwelligkeit vorerst vereinfacht, eine präzise Visualisierung wird in Abschnitt 1.2.2 besprochen.

Die Amplituden sind komplexe Zahlen (Infobox 2), welche die Normierungsbedingung  $|\alpha|^2 + |\beta|^2 = 1$  erfüllen müssen. Wir werden nachfolgend im Text noch genauer erklären welche Bedeutung  $\alpha$  und  $\beta$  haben und was die Normierungsbedingung aussagt. Vorerst reicht uns allerdings die Vorstellung, dass uns  $\alpha$  und  $\beta$  sagen zu welchem Anteil die Basiszustände  $|0\rangle$  bzw.  $|1\rangle$  zu dem Superpositionszustand  $|\psi\rangle$  beitragen (siehe **Abbildung 1** zur vereinfachten Vorstellung eines Qubit-Zustandes und Vergleich zum klassischen Bit).

#### Infobox 2: Komplexe Zahlen

Eine komplexe Zahl  $z$  ist definiert durch zwei reelle Zahlen  $x$  und  $y$  gemäß  $z = x + iy$ . Hierbei ist  $i = \sqrt{-1}$  die imaginäre Einheit und reelle Zahlen sind die mathematische Bezeichnung für die Zahlen, mit denen wir üblicherweise rechnen, wie z. B.  $0, 1, 2, -1, -10, 2.5, \frac{2}{3}, \pi, \sqrt{2}, \dots$

Der Betrag einer komplexen Zahl berechnet sich durch  $|z| = \sqrt{x^2 + y^2}$ .

### 1.2.2 Darstellung auf der Bloch-Sphäre

Wir haben oben gesehen, dass jedes Qubit  $|\psi\rangle$  über zwei Amplituden definiert ist. Diese wiederum sind komplexe Zahlen und somit jeweils durch zwei reelle Zahlen definiert. Dies würde  $2 \cdot 2 = 4$  Freiheitsgrade für ein Qubit ergeben. Einen Freiheitsgrad verlieren wir jedoch durch die Normierungsbedingung. Zudem stellt sich heraus, dass Qubits, welche sich nur um eine sogenannte globale Phase (Infobox 3) unterscheiden, in Messungen nicht unterscheidbar sind.

### Infobox 3: Globale Phase

Zwei Qubits  $|\psi_1\rangle$  und  $|\psi_2\rangle$  unterscheiden sich nur in einer globalen Phase, falls  $|\psi_1\rangle = \gamma |\psi_2\rangle$  für eine komplexe Zahl  $\gamma$  mit  $|\gamma|=1$  gilt. Z. B.  $|\psi_1\rangle = (-1) \cdot |\psi_2\rangle$ .

Diese Qubits werden entsprechend miteinander identifiziert, was einen weiteren Freiheitsgrad kostet. Es bleiben somit zwei Freiheitsgrade für das Qubit  $|\psi\rangle$  übrig. Diese werden oft in Breiten- und Längengradkoordinaten übersetzt, welche mit zwei Winkeln  $\theta$  (für den Breitengrad) und  $\phi$  (für den Längengrad) bezeichnet werden. Damit lässt sich das Qubit  $|\psi\rangle$  als Punkt auf der Oberfläche einer Kugel, der sogenannten **Bloch-Sphäre**, visualisieren. In **Abbildung 2** ist dies dargestellt.

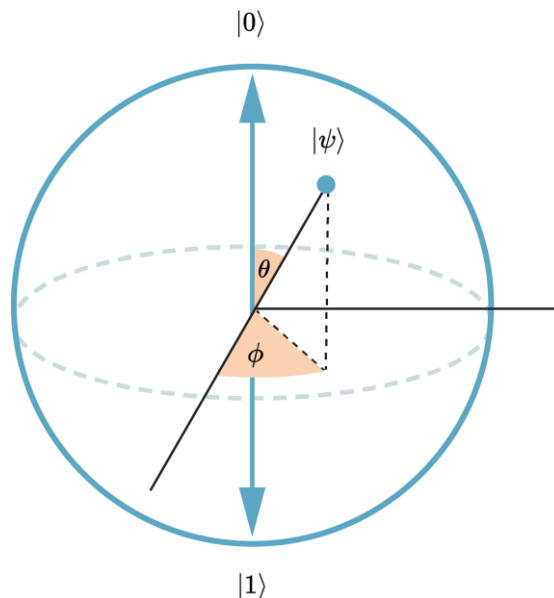


Abbildung 2: Darstellung des Qubits  $|\psi\rangle$  auf der Bloch-Sphäre

Auf der Bloch-Sphäre sind die Basiszustände  $|0\rangle$  und  $|1\rangle$  der Nord- bzw. Südpol der Sphäre. So weit wäre man noch beim klassischen Bit mit seinen binären Zuständen 0 und 1. Auf der Bloch-Sphäre gibt es jedoch noch unendlich viele weitere Punkte zwischen dem Nord- und Südpol. Jeder davon entspricht einer anderen Superposition aus  $|0\rangle$  und  $|1\rangle$ , und ist damit insbesondere ein zulässiger Zustand für unser Qubit  $|\psi\rangle$ . Diese Superpositionszustände sind nur auf dem Quantencomputer möglich, es gibt nichts Gleichwertiges in der klassischen Computertheorie. Sie sind eine der Ressourcen, aus denen sich ein Effizienzvorteil gegenüber dem klassischen Computer gewinnen lässt.



## 1.2.3 Qubits messen

### 1.2.3.1 Allgemeine Konzepte

Wollen wir etwas über den Zustand eines Qubit erfahren, müssen wir es messen. Eine solche Messung kann jedoch ausschließlich klassische Information liefern. Genauer: die Messung eines Qubits liefert ein Bit Information, kann also nur 0 oder 1 als Ergebnis liefern. Ob das Messresultat dabei 0 oder 1 ist, kann nicht vorhergesagt werden. Jedoch können die Wahrscheinlichkeiten beider Ausgänge charakterisiert werden. Diese Information steckt in den Amplituden  $\alpha$  und  $\beta$ . Genauer gilt: die Wahrscheinlichkeit, 0 als Messergebnis zu erhalten ist gegeben durch  $|\alpha|^2$  und die Wahrscheinlichkeit, eine 1 zu messen durch  $|\beta|^2$ . In diesem Zusammenhang können wir auch die obige Normierungsbedingung  $|\alpha|^2 + |\beta|^2 = 1$  verstehen: Sie garantiert, dass die Wahrscheinlichkeiten, eine 0 oder eine 1 zu messen sich immer zu 1 (also zu 100%) addieren. In Infobox 4 haben wir drei Beispiele zur Messung von Qubits zusammengestellt.

### 1.2.3.2 Zusammenhang mit der Bloch-Sphäre

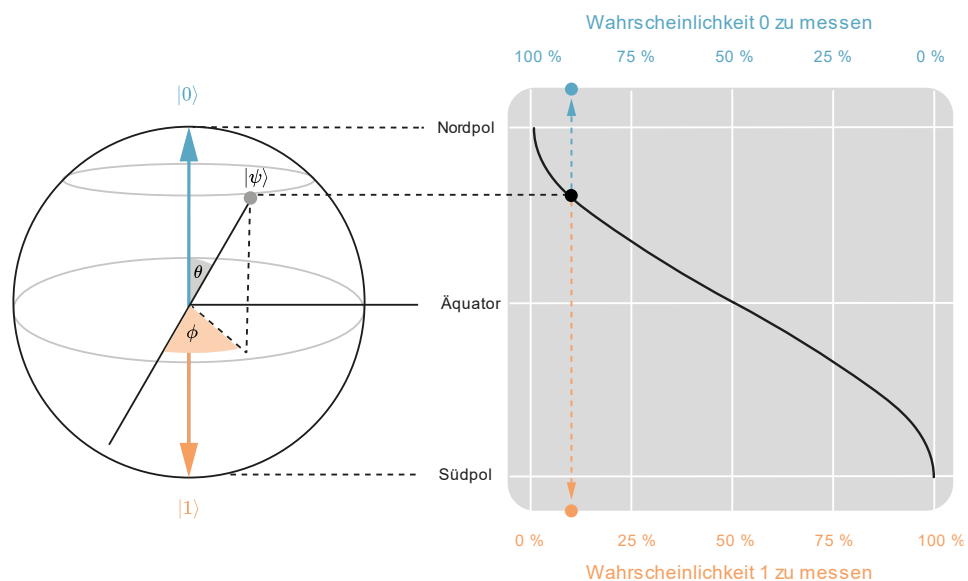


Abbildung 3: Die Messung eines Qubits kann nur 0 oder 1 zum Ergebnis haben. Der Breitengradwinkel  $\theta$  codiert mit welcher Wahrscheinlichkeit welches Ergebnis auftreten wird. Je nördlicher sich das Qubit  $|\psi\rangle$  befindet desto wahrscheinlicher ist 0, je südlicher es sich befindet desto wahrscheinlicher wird 1. Der Längengradwinkel  $\phi$  spielt für die Messung keine Rolle.

Auf der Bloch-Sphäre sind die Messwahrscheinlichkeiten im Breitengrad  $\theta$  codiert, siehe **Abbildung 3**. Befindet sich  $|\psi\rangle$  auf dem Äquator, erhalten wir mit 50 % Wahrscheinlichkeit 0 und mit 50 % Wahrscheinlichkeit 1 bei der Messung. Ist  $|\psi\rangle$  nördlich vom Äquator so ist die Wahrscheinlichkeit, eine 0 zu messen größer als 50 % (und eine 1 zu messen

entsprechend geringer als 50 %). Dabei gilt, dass je weiter nördlich  $|\psi\rangle$  ist, desto größer wird die Wahrscheinlichkeit, eine 0 zu messen, bis sie am Nordpol 100 % ist. Genau andersherum ist es, wenn sich  $|\psi\rangle$  auf der Südhalbkugel befindet. Hier wächst die Wahrscheinlichkeit, eine 1 zu messen, bis sie 100 % am Südpol beträgt. Mathematisch lässt sich dies präzise fassen: die Wahrscheinlichkeit, eine 0 zu messen ist gegeben als  $\cos^2 \theta$  und die Wahrscheinlichkeit, eine 1 zu messen als  $\sin^2 \theta$ . Aus den trigonometrischen Additionstheoremen wissen wir, dass für jeden Winkel  $\theta$  die Beziehung  $\cos^2 \theta + \sin^2 \theta = 1$  gilt und somit auch in dieser Darstellung die Normierungsbedingung erfüllt ist. Bemerkenswerterweise spielt der Längengradwinkel  $\Phi$  für den Ausgang der Messung keine Rolle.

#### Infobox 4: Qubits messen: Beispiele

Wir betrachten drei verschiedene Qubit-Zustände wie sie in Tabelle 1 beschrieben sind. Bei (a) haben wir eine gleichmäßige Superposition von  $|0\rangle$  und  $|1\rangle$ , d.h. beide Amplituden haben den gleichen Betrag. Entsprechend sehen wir, dass sich die Wahrscheinlichkeiten 0 bzw. 1 zu messen sich 50 % zu 50 % verteilen. Beispiel (b) hat eine betragsmäßig größere Amplitude für den Basiszustand  $|0\rangle$  als für den Basiszustand  $|1\rangle$ . Entsprechend ist die Wahrscheinlichkeit 0 bei einer Messung zu erhalten größer als die 1 zu erhalten (in diesem konkreten Beispiel 75 % zu 25 %). Zuletzt hat (c) die gleiche Amplitude für  $|0\rangle$  wie (b), die Amplitude für  $|1\rangle$  unterscheidet sich jedoch ( $1/2$  für (b),  $i/2$  für (c)). Betragsmäßig sind sie jedoch gleich groß, sodass auch die Messwahrscheinlichkeiten von (b) und (c) gleich sind.

Zustände (b) und (c) unterscheiden sich durch eine sogenannte relative Phase. Diese ist nicht zu verwechseln mit der globalen Phase (Infobox 3; Zustände (b) und (c) sind keine Vielfachen voneinander), denn sie ist physikalisch erfassbar und spielt eine wichtige Rolle in vielen Quanten-Algorithmen.

	Qubit $\alpha 0\rangle + \beta 1\rangle$	$ \alpha ^2$	Wahrscheinlichkeit 0 zu messen $ \alpha ^2 \cdot 100\%$	$ \beta ^2$	Wahrscheinlichkeit 1 zu messen $ \beta ^2 \cdot 100\%$
(a)	$\frac{1}{\sqrt{2}} 0\rangle + \frac{1}{\sqrt{2}} 1\rangle$	$\frac{1}{2}$	$\frac{1}{2} \cdot 100\% = 50\%$	$\frac{1}{2}$	$\frac{1}{2} \cdot 100\% = 50\%$
(b)	$\frac{\sqrt{3}}{2} 0\rangle + \frac{1}{2} 1\rangle$	$\frac{3}{4}$	$\frac{3}{4} \cdot 100\% = 75\%$	$\frac{1}{4}$	$\frac{1}{4} \cdot 100\% = 25\%$
(c)	$\frac{\sqrt{3}}{2} 0\rangle + \frac{i}{2} 1\rangle$	$\frac{3}{4}$	$\frac{3}{4} \cdot 100\% = 75\%$	$\frac{1}{4}$	$\frac{1}{4} \cdot 100\% = 25\%$

Tabelle 1: Beispiele für unterschiedliche Qubit-Zustände mit zugehörigen Wahrscheinlichkeiten bei einer Messung das Ergebnis 0 oder 1 zu erhalten.

Dieser codiert die in Infobox 4 angesprochene relative Phase zwischen  $|0\rangle$  und  $|1\rangle$  und ist deshalb für den Ausgang der Messung unerheblich. Wir wollen jedoch anmerken, dass sie

für viele Quantenalgorithmen eine große Rolle spielt. In Infobox 5 haben wir nochmals die Beispiele aus Infobox 4 in ihrer Version auf der Bloch-Sphäre beschrieben.

#### Infobox 5: Qubits messen: Beispiele (Bloch-Sphäre-Version)

In **Abbildung 4** und **Tabelle 2** haben wir die drei Beispiele **Tabelle 1** in ihrer Repräsentation auf der Bloch-Sphäre dargestellt. Wir sehen, dass der Ausgang einer Messung nur vom Breitengrad abhängt, auf dem sich das Qubit befindet. Am Äquator haben wir sowohl 50 % Wahrscheinlichkeit als Messergebnis 0 oder 1 zu erhalten. Je weiter nördlich der Breitengrad vom Äquator ist, desto wahrscheinlicher wird 0, und andersherum je weiter südlich desto wahrscheinlicher wird 1.

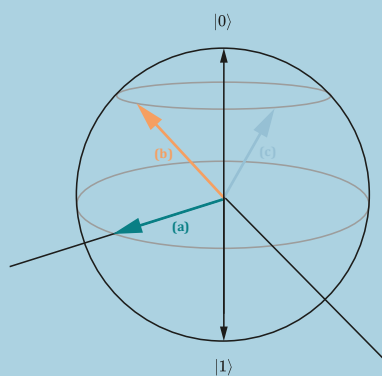


Abbildung 4: Beispiele (a), (b) und (c) visualisiert auf der Bloch-Sphäre.

	Qubit $\alpha 0\rangle + \beta 1\rangle$	Winkel $\theta$	Winkel $\varphi$	Wahrscheinlichkeit 0 zu messen	Wahrscheinlichkeit 1 zu messen
(a)	$\frac{1}{\sqrt{2}} 0\rangle + \frac{1}{\sqrt{2}} 1\rangle$	$90^\circ$	$0^\circ$	50 %	50 %
(b)	$\frac{\sqrt{3}}{2} 0\rangle + \frac{1}{2} 1\rangle$	$60^\circ$	$0^\circ$	75 %	25 %
(c)	$\frac{\sqrt{3}}{2} 0\rangle + \frac{i}{2} 1\rangle$	$60^\circ$	$90^\circ$	75 %	25 %

### 1.2.3.3 Kollaps der Wellenfunktion

Wir haben gesehen, dass der Ausgang einer Messung des Qubits  $|\psi\rangle$  probabilistisch ist. Mit Wahrscheinlichkeit  $|\alpha|^2$  messen wir 0 und mit Wahrscheinlichkeit  $|\beta|^2$  messen wir 1. Misst man nun das Qubit ein weiteres Mal ist das Ergebnis jedoch deterministisch: Haben wir bei der ersten Messung 0 gemessen, wird jede weitere Messung 0 liefern und haben wir 1 gemessen wird das Ergebnis jeder weiteren Messung 1 sein. Der Grund dafür ist,

dass das Messen des Qubits dessen Zustand  $|\psi\rangle$  geändert hat. Genauer gesagt, ändert sich sein Zustand zu  $|0\rangle$ , falls das Messresultat 0 war, und zu  $|1\rangle$ , falls es 1 war, siehe **Abbildung 5**. Dies wird als **Kollaps** der Wellenfunktion bezeichnet.

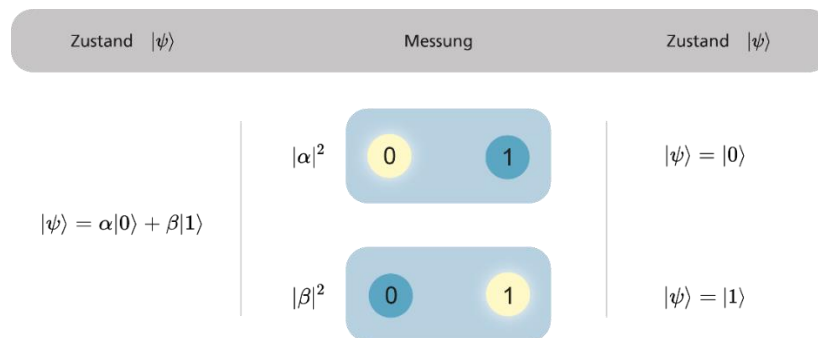


Abbildung 5: Durch die Messung kollabiert der Superpositionszustand von  $|\psi\rangle$ . Daraus ergibt sich die weitreichende Konsequenz, dass **das Messen eines Qubits den Zustand desselbigen ändert** und somit nicht zwischen Rechenoperationen angewandt werden kann, ohne den Ausgang der Gesamtrechnung zu beeinflussen. Im Unterschied zum klassischen Computer ist eine Abfolge von Rechenoperationen auf dem Quantencomputer (d.h. ein Quantenalgorithmus) also eine **Black Box**. Wir können nur Daten eingeben und das finale Ergebnis messen, die Zwischenschritte können nicht ausgelesen oder überwacht werden. Dies hat weitreichende Konsequenzen für das Software-Engineering, z. B. im Test und der Validierung von Quantensoftware, siehe Kapitel 3.1.2.4.

### Infobox 6: Analogie: Münzwurf

Betrachten wir das Qubit  $|\psi\rangle$  in der gleichmäßigen Superposition  $|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ , in welcher die beiden Basiszustände  $|0\rangle$  und  $|1\rangle$  mit gleicher Amplitude vertreten sind (auf der Bloch-Sphäre würde sich das Qubit auf dem Äquator befinden). Bei einer Messung erhalten wir jeweils mit Wahrscheinlichkeit 50 % das Ergebnis 0 oder 1. Durch die Messung kollabiert die Superposition, sodass im Anschluss das Qubit in Zustand  $|0\rangle$  bzw.  $|1\rangle$  ist. Als grobe Analogie dafür können wir uns eine Münze vorstellen, auf welcher die Seiten mit 0 und 1 beschriftet sind. Werfen wir diese Münze in die Luft, fangen sie auf und schauen uns anschließend an welche Seite oben liegt, haben wir auch hier 50 % Wahrscheinlichkeit, dass wir 0 bzw. 1 sehen. Aber solange die Münze in der Luft war, waren beide Ausgänge noch möglich und insbesondere gleich wahrscheinlich. Die Münze befand sich also salopp gesprochen in einer analogen Superposition wie unser Qubit  $|\psi\rangle$  vor der Messung. Nachdem wir die Münze allerdings gefangen haben (gemessen haben), steht die oben liegende Seite fest – die Superposition ist kollabiert.

## 1.2.4 Multi-Qubit-Systeme

Bei einem System aus zwei klassischen Bits kann jedes Bit unabhängig von dem anderen zwei mögliche Zustände annehmen. Das ergibt insgesamt  $2 \cdot 2 = 2^2 = 4$  mögliche Zustände für das System: 00, 01, 10 und 11. Analog hat ein System aus zwei Qubits  $2^2 = 4$  Basiszustände, welche mit  $|00\rangle$ ,  $|01\rangle$ ,  $|10\rangle$  und  $|11\rangle$  bezeichnet werden. Wie im Fall eines einzelnen Qubits ist jedoch jede Superposition, die die Normierungsbedingung  $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$  erfüllt, ein gültiger Zustand für ein solches Zwei-Qubit-System. Für ein System aus drei Qubits haben wir  $2 \cdot 2 \cdot 2 = 2^3 = 8$  Basiszustände  $|000\rangle$ ,  $|001\rangle$ ,  $|010\rangle$ ,  $|011\rangle$ ,  $|100\rangle$ ,  $|101\rangle$ ,  $|110\rangle$  und  $|111\rangle$  mit entsprechend vielen Amplituden. Allgemein kann der Zustand eines Systems aus  $n$  Qubits durch  $2^n$  Basiszustände und deren zugehörigen Amplituden als

$$|\psi\rangle = \alpha_{00\dots00}|00\dots00\rangle + \alpha_{00\dots01}|00\dots01\rangle + \dots + \alpha_{11\dots10}|11\dots10\rangle + \alpha_{11\dots11}|11\dots11\rangle$$

mit entsprechender Normierungsbedingung beschrieben werden. Wir sehen, dass die Anzahl der benötigten Amplituden sich mit jedem Qubit verdoppelt ( $n$  Qubits benötigen  $2^n$  Amplituden wohingegen  $n+1$  Qubits  $2^{n+1}$  Amplituden benötigen. Es gilt  $2^{n+1} = 2 \cdot 2^n$ , woraus wir die angesprochene Verdopplung ablesen können). Dieses exponentielle Wachstum ist in Tabelle 3 dargestellt und dem Speicherbedarf, welcher auf einem klassischen Computer für ein solches  $n$ -Qubit-System erforderlich wäre, gegenübergestellt. Zur besseren Einschätzung wollen wir erwähnen, dass ein gängiger Computer 8 bis 16 Gigabyte Arbeitsspeicher besitzt, während der aktuell größte Supercomputer der Welt 9,2 Petabyte (=9,2 Millionen Gigabyte) an Arbeitsspeicher hat und für seinen Betrieb einen elektrischen Leistungsbedarf von bis zu 21.000 kW [G 5], [G 6], erfordert (zum Vergleich ein zwölfteiliger ICE 4 hat 9.900 kW Antriebsleistung [G 7]). Die Simulation eines Quantencomputers auf einem klassischen Computer erfordert also bereits bei einer moderaten Anzahl von Qubits Höchstleistungsrechner und selbst diese erreichen schnell die Grenze des Machbaren. Auf den derzeit größten Hochleistungsrechnern können maximal 45 Qubits exakt und 48 Qubits mit guter Approximation simuliert werden [G 8].

Anzahl Qubits	Anzahl möglicher Zustände	Benötigter Arbeitsspeicher
1	2	16 Byte
2	4	32 Byte
3	8	64 Byte
5	32	256 Byte

10	1.024	> 16 Kilobyte
20	> 1.000.000	> 16 Megabyte
30	> 1.000.000.000	> 16 Gigabyte
40	> 1.000.000.000.000	> 16 Terabyte
50	> 1.000.000.000.000.000	> 16 Petabyte
300	> Anzahl Atome im Universum	-

Tabelle 3: Anzahl möglicher Zustände für Qubit-Systeme und benötigter Speicher auf einem klassischen Computer.

### 1.2.4.1 Quantenparallelismus

Auch mit einem klassischen Computer mit  $n$  Bits kann man  $2^n$  Zustände darstellen (z.B. die Zahlen von 0 bis  $2^n - 1$ ). Will man eine gegebene Rechenoperation für alle diese Zustände ausführen, muss der klassische Computer jeden der Zustände einzeln und nacheinander manipulieren (sprich: die Rechenoperation anwenden). Da es  $2^n$  unterschiedliche Zustände gibt, muss der klassische Computer also  $2^n$  Rechenoperationen ausführen. Hier liegt demnach ein exponentielles Komplexitätswachstum vor: Die Anzahl der Rechenoperationen verdoppelt sich, wenn ein weiteres Bit hinzukommt. Eine Besonderheit am Quantencomputer ist, dass ein System aus  $n$  Qubits alle  $2^n$  Zustände simultan in einer einzigen Superposition darstellen kann und diese mit **nur einer Rechenoperation alle gleichzeitig manipulieren** kann. Wächst die Anzahl der Qubits, bleibt es dennoch stets bei einer Rechenoperation. Dieser Effekt ist als **Quantenparallelismus** bekannt und eine der Quellen für die Mächtigkeit von Quantencomputern. In Infobox 7 haben wir den Effekt des Quantenparallelismus am Beispiel eines einfachen Sudokus veranschaulicht.

Allerdings gibt es eine große Herausforderung, um Nutzen aus dem Quantenparallelismus zu gewinnen. Wir können zwar bei einer Superposition aus  $n$  Qubits auf  $2^n$  Zuständen parallel rechnen, um das Resultat aber auszulesen, müssen wir die Qubits messen. Aus einer Messung erhalten wir jedoch nur einen der  $2^n$  Zustände als Messergebnis, und zwar mit einer Wahrscheinlichkeit, die in den Amplituden codiert ist. Zusätzlich kollabiert die Superposition, wodurch eine Wiederholung des Messprozesses keine weitere Information bereitstellt. Wir haben also mit dem Quantenparallelismus durch eine Rechenoperation ein Quantenzustand mit allen Resultaten erzeugt, können jedoch nicht ohne Weiteres auf sie zugreifen. Eine große Herausforderung des Designs von Quantenalgorithmen ist demnach die Amplituden der erwünschten Zustände zu verstärken, sodass die Wahrscheinlichkeit diese bei der Messung zu erhalten hoch ist.

### Infobox 7: Quantenparallelismus am Sudoku-Beispiel

Wir betrachten ein 2 x 2 Sudoku mit vier Einträgen:

0/1	0/1
0/1	0/1

Hieraus lassen sich  $2^4 = 16$  verschiedene Kombinationen bilden, wovon zwei ein gültiges Sudoku-Board ergeben.

Möchten wir die Gültigkeit jeder dieser Kombinationen auf dem klassischen Computer herausfinden, bleibt nichts anderes übrig als nacheinander jede einzelne Kombination zu überprüfen. Insgesamt sind also 16 Rechnungen auszuführen.

Rechnung Nummer	Kombination und Ergebnis				
1	<table border="1"> <tr><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td></tr> </table> <span style="color: red;">✗</span>	0	0	0	0
0	0				
0	0				
2	<table border="1"> <tr><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td></tr> </table> <span style="color: red;">✗</span>	1	0	0	0
1	0				
0	0				
...	...				
7	<table border="1"> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table> <span style="color: green;">✓</span>	0	1	1	0
0	1				
1	0				
...	...				
16	<table border="1"> <tr><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td></tr> </table> <span style="color: red;">✗</span>	1	1	1	1
1	1				
1	1				

Ganz anders auf dem Quantencomputer: Hier können wir mit vier Qubits alle 16 Kombinationen in einer einzigen Superposition darstellen und simultan auf ihre Gültigkeit überprüfen.

Rechnung Nummer	Kombination und Ergebnis																
1	<table border="1"> <tr><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td></tr> </table> <span style="color: red;">✗</span> <table border="1"> <tr><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td></tr> </table> <span style="color: red;">✗</span> <table border="1"> <tr><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td></tr> </table> <span style="color: red;">✗</span> <table border="1"> <tr><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td></tr> </table> <span style="color: red;">✗</span>	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0
0	0																
0	0																
1	0																
0	0																
0	1																
0	0																
1	1																
0	0																
	<table border="1"> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td></tr> </table> <span style="color: red;">✗</span> <table border="1"> <tr><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td></tr> </table> <span style="color: red;">✗</span> <table border="1"> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table> <span style="color: green;">✓</span> <table border="1"> <tr><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table> <span style="color: red;">✗</span>	0	0	1	0	1	0	1	0	0	1	1	0	1	1	1	0
0	0																
1	0																
1	0																
1	0																
0	1																
1	0																
1	1																
1	0																
	<table border="1"> <tr><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td></tr> </table> <span style="color: red;">✗</span> <table border="1"> <tr><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td></tr> </table> <span style="color: green;">✓</span> <table border="1"> <tr><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td></tr> </table> <span style="color: red;">✗</span> <table border="1"> <tr><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td></tr> </table> <span style="color: red;">✗</span>	0	0	0	1	1	0	0	1	0	1	0	1	1	1	0	1
0	0																
0	1																
1	0																
0	1																
0	1																
0	1																
1	1																
0	1																
	<table border="1"> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </table> <span style="color: red;">✗</span> <table border="1"> <tr><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </table> <span style="color: red;">✗</span> <table border="1"> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td></tr> </table> <span style="color: red;">✗</span> <table border="1"> <tr><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td></tr> </table> <span style="color: red;">✗</span>	0	0	1	1	1	0	1	1	0	1	1	1	1	1	1	1
0	0																
1	1																
1	0																
1	1																
0	1																
1	1																
1	1																
1	1																

Autor: Dr. Andreas Sturm (Fraunhofer IAO)

Kontakt: [andreas.sturm@iao.fraunhofer.de](mailto:andreas.sturm@iao.fraunhofer.de)

### 1.3 Hardwarerealisierungen von Quantencomputern

Um die grundlegende Physik verschiedener Quantencomputerarchitekturen nachvollziehen zu können, ist es unumgänglich zu verstehen: Was sind eigentlich »Quanten«? Diese grundlegende Frage soll im Folgenden insbesondere für Nicht-Naturwissenschaftler erläutert werden.

Kurz gesagt versteht man unter dem Begriff »Quanten« infinitesimale (sehr kleine!) und diskrete Energieanregungen eines beliebigen Systems - Eine Definition, die zugegebenermaßen für die meisten Zielgruppen jedoch völlig unverständlich und damit nutzlos erscheint. Obgleich jedes noch so kleine Detail unserer Welt auf Quanten basiert, können wir diese mit dem bloßen Auge nicht erkennen. Dies macht die physikalischen Prinzipien der Quantenmechanik für uns so schwer nachvollziehbar. Um den Begriff des „Quants“ dennoch verständlich zu machen, bedienen wir uns im Folgenden eines wohlbekannten Beispiels aus dem ursprünglichen Navigationsbereich: Der Kompassnadel, ein Stabmagnet mit magnetischem Nord- und Südpol.

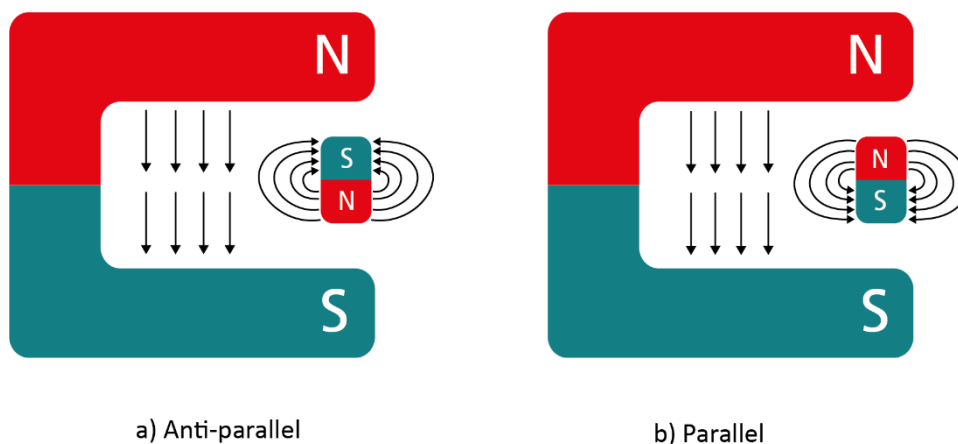


Abbildung 6: Stabmagnet (Kompassnadel) im homogenen Feld eines Hufeisenmagneten. Die magnetischen Nord- und Südpole sind mit N (rot) und S (grün) gekennzeichnet. Die Teilabbildungen a) bzw. b) definieren die energetisch niedrigsten und höchsten Ausrichtungen des Stabmagnetes im homogenen externen Feld: a) Anti-parallele Ausrichtung der Magnetfeldlinien (Grundzustand) bzw. b) parallele Ausrichtung der Magnetfeldlinien (angeregter Zustand).

In unserem Gedankenexperiment legen wir einen Stabmagneten (Kompassnadel) in das homogene Feld eines Hufeisenmagneten, wie in **Abbildung 6** gezeigt. Einmal dem externen Magnetfeld ausgesetzt, ist es wohlbekannt, dass der Stabmagnet sich exakt so ausrichtet, dass seine Magnetfeldlinien zu denen des externen Feldes anti-parallel stehen. Dieser energetische Grundzustand des Systems ist in **Abbildung 6a**) dargestellt. Weicht die Lage des Stabmagneten von diesem Zustand nur leicht ab, so wirkt auf ihn ein endliches Drehmoment und der Stab wird wieder zurück in die Ausgangsposition gezwungen. Es gibt jedoch genau einen Zustand, bei dem dies nicht der Fall ist, genau



dann, wenn die Feldlinien des Stabmagneten und die des Hufeisenmagneten exakt parallel zueinanderstehen. Dies ist in **Abbildung 6b)** dargestellt. Obgleich jene Konfiguration der Zustand höchster Energie ist, wirkt in dieser Position kein Drehmoment und das System ist in einem sogenannten instabilen Gleichgewicht.

Salopp gesprochen, kann man anhand dieses Gedankenexperiments das Prinzip des „Quants“ erläutern. Nehmen wir an, das obige System hätte nur die beiden erlaubten Konfigurationen (a) anti-parallel und (b) parallel mit der positiven Energiedifferenz  $\Delta E = E_{\downarrow\downarrow} - E_{\downarrow\uparrow}$ . Befindet sich der Stabmagnet im Grundzustand (anti-parallel) und regen wir das System genau mit der Energie  $\Delta E$  an, so absorbiert der Stabmagnet dieses „Energiequant“ und wechselt in den parallelen, angeregten Zustand. Dieser fiktive Prozess veranschaulicht den zentralen Mechanismus aller Quantencomputerarchitekturen. Mittels der Absorption bzw. der Emission von Energiepaketen werden in einem Quantencomputer Übergänge zwischen den Grundzuständen  $|0\rangle$  sowie den angeregten Zuständen  $|1\rangle$  induziert. Dieser fundamentale und für den weiteren Fortgang des Kapitels unverzichtbare Prozess ist in **Abbildung 7** visualisiert.

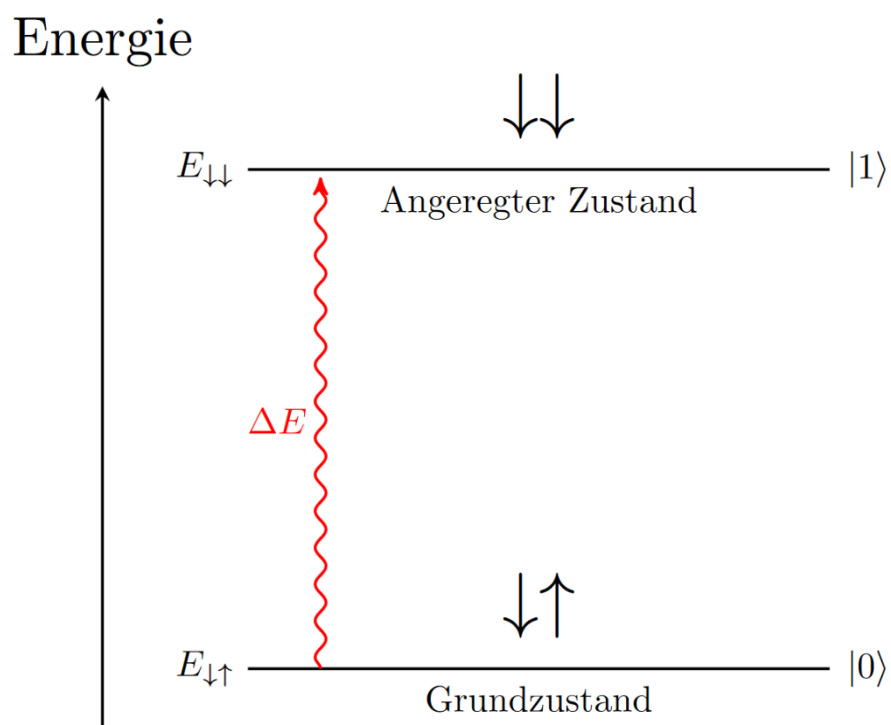


Abbildung 7: Energetischer Grund-  $|0\rangle$  und angeregter Zustand  $|1\rangle$  eines Quantensystems, die mittels Emission- bzw. Absorption eines Energiequants (rote gewellte Linie) ineinander überführt werden können. Derartige Übergänge werden auf einem Quantencomputer durch sogenannte Gates induziert. Beispiel: Vom Grundzustand  $|0\rangle$  in den angeregten Zustand  $|1\rangle$  eines Quantensystems gelangt man durch das Anwenden eines X-Gates.

## Künstliche Atome und das Zwei-Level-System

Bei genauer Betrachtung hinkt die obige Analogie zwischen Stabmagneten und Quantensystem natürlich. Strenggenommen ist es dem Stabmagneten möglich, jede Position im externen Magnetfeld anzunehmen. Daher sind seine positionsabhängigen Energiewerte kontinuierlich und nicht auf zwei diskrete Level beschränkt. Ein echtes Quantensystem ist beispielsweise das Wasserstoffatom, welches man eventuell noch aus der Schulzeit kennt. In diesem Atom umkreist ein negativ geladenes Elektron ein positiv geladenes Proton. Dabei kann das Elektron in verschiedenen sogenannten Orbitalen sitzen, wie strukturell in **Abbildung 8** dargestellt. Jedes Orbital besitzt einen diskreten Energiewert und Übergänge können mittels Absorption bzw. Emission von Energiequanten erzeugt werden.

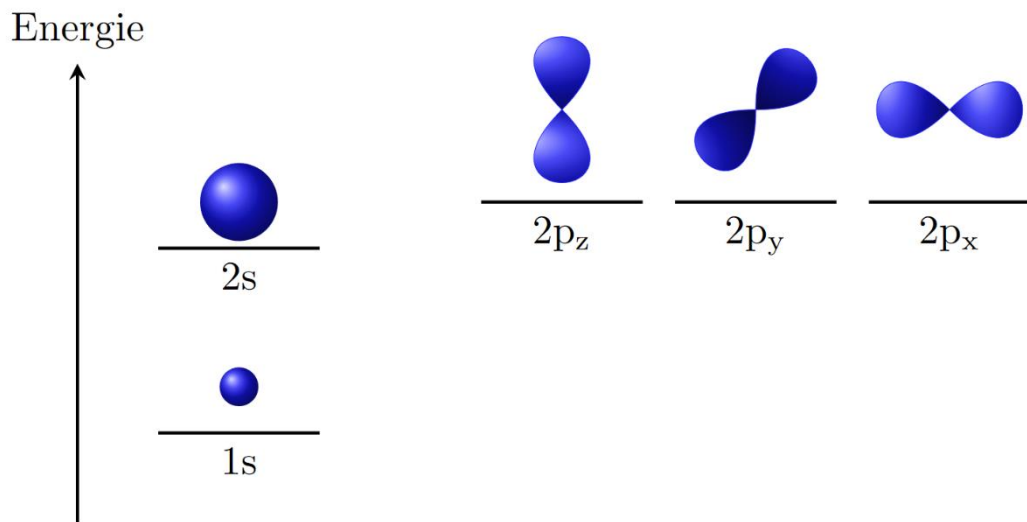


Abbildung 8: Kugel- (s) und Hantelatomorbitale (p) sowie zugehörige Energieniveaus eines Elektrons in einem Atom (strukturell für die Hauptquantenzahlen  $n = 1$  und  $n = 2$ ). Übergänge im Spektrum können mittels Emission bzw. Absorption von Energiequanten, z.B. in Form von Licht (elektromagnetischen Wellen) erzeugt werden. Bei Energieniveaus gleicher Energie spricht man von Entartung. Die p-Orbitale unterscheiden sich in der Quantenzahl für den Bahndrehimpuls des Elektrons. Im einfachsten Modell sind diese entartet, wobei die Entartung aufgespalten wird, wenn weitere Effekte wie Spin-Bahn Kopplung oder eine Wechselwirkung mit einem äußeren elektrischen oder magnetischen Feld vorliegen.

Aufgrund der diskreten Struktur Ihrer Energielevel können also Atome in natürlicher Art und Weise zum Quantencomputing genutzt werden. Zur Konstruktion eines Quantencomputers benötigt man (zumindest für Qubits) jedoch nur zwei der oben genannten diskreten Energieniveaus, einen Grundzustand  $|0\rangle$  und einen angeregten Zustand  $|1\rangle$ . Welche Niveaus man hier konkret wählt hängt von zahlreichen Randbedingungen ab und kann im Rahmen dieser Studie nicht im Detail vertieft werden. Um dennoch Ideen für jene Auswahl zu vermitteln, soll an dieser Stelle angemerkt werden, dass ähnlich zum radioaktiven Zerfall in Atomkraftwerken, Anregungen eines

jeden Energieniveaus nur eine endliche Lebensdauer besitzen. D.h. insbesondere, dass Informationen, die im Zustand  $|1\rangle$  gespeichert wurden, nach einer sogenannten „Kohärenzzeit“ verloren gehen. Dies ist eine signifikante Fehlerquelle für heutige Quantencomputer, deren Konsequenz eingehend in Kapitel 1.5 diskutiert wird. Ein weiterer prominenter Fehler im Quantencomputing resultiert daraus, dass der Anregungsimpuls  $|0\rangle \rightarrow |1\rangle$  ebenfalls höher liegende Energieniveaus  $|2\rangle, |3\rangle, \dots$  anregt, die jedes Quantensystem unweigerlich besitzt (Ansatzpunkt für Qudit-Technologien von Quantencomputern [H 1]).

Neben „echten“ Atomen, Ionen oder Molekülen, die ein natürliches diskretes Energiespektrum besitzen, ist es heutzutage Wissenschaftlern auch möglich sogenannte „künstliche“ Atome zu konstruieren. Unter künstlichen Atomen versteht man elektrische Schaltkreise (eine Aneinanderreihung elektronischer Bauteile) im Nano- bis Mikrometerbereich, die analog zu „echten“ Atomen ein diskretes Energiespektrum besitzen und so ebenfalls zum Quantencomputing genutzt werden können. Beispielsweise fallen supraleitende Qubits und Quanten-Punkte in die Kategorie künstlicher Atome, während Ionenfallen und NV-Zentren als atomähnliche Systeme zu kategorisieren sind. Im Folgenden werden wir auf die wichtigsten Prinzipien jener Technologien, sowie deren charakteristische Vor- und Nachteile eingehen.

Autoren: Dr. Christian Tutschku (Fraunhofer IAO), Daniel Pranjic (Fraunhofer IAO)

Kontakt: [christian.tutschku@iao.fraunhofer.de](mailto:christian.tutschku@iao.fraunhofer.de), [daniel.pranjic@iao.fraunhofer.de](mailto:daniel.pranjic@iao.fraunhofer.de)

### 1.3.1 Ionenfallen Technologie

Wir beginnen unsere Reise durch die heutigen Quantencomputingtechnologien mit dem sogenannten Ionenfallenansatz. Per Definition sind Ionen elektrisch geladene Atome oder Moleküle. Im ungeladenen Zustand sind Atome oder Moleküle aus gleich vielen Elektronen und Protonen aufgebaut. Ändert man die Elektronenkonfiguration jenes neutralen Zustandes, so erhält man positiv bzw. negativ geladene Ionen, je nachdem ob man bzgl. des ungeladenen Zustandes Elektronen hinzugefügt oder entfernt hat. Als geladene Teilchen können Ionen mittels elektrischer oder magnetischer Felder manipuliert werden, so wie es beispielsweise mit geladenen Teilchen im Erdmagnetfeld der Fall ist – ein konkretes Phänomen wären Polarlichter.

Gleichnamige elektrische Ladungen stoßen sich aufgrund ihrer entgegengesetzten elektrischen Felder ab, während sich ungleichnamige elektrische Ladungen anziehen. Physikalisch ist die Kraft, die ein geladenes Teilchen in einem elektromagnetischen Feld erfährt durch die sogenannte Lorenz-Kraft gegeben – eine Kraft, die uns mittels der

charakteristischen „Drei-Finger Regel“ noch aus der schulischen Ausbildung bekannt ist. Unter einer Ionenfalle versteht man nun das Konstruieren einer geeigneten elektromagnetischen Feldkonfiguration, die auf diesem Effekt basierend ein Ion an genau einer Stelle im Raum fixiert. Während genau an dieser Stelle die elektromagnetische Nettokraft auf das Ion verschwindet, wird das geladene Teilchen nach jeder Auslenkung durch rücktreibende Kräfte wieder auf den Gleichgewichtspunkt zurückgeführt. Eine derartige Apparatur entwickelte der deutsche Physiker Wolfgang Paul im Jahre 1983, die sogenannte „Paul-Falle“, für die er 1989 mit dem Nobelpreis ausgezeichnet wurde (gemeinsam mit Hans Georg Dehmelt für dessen Forschung an der Penning-Falle 1987).

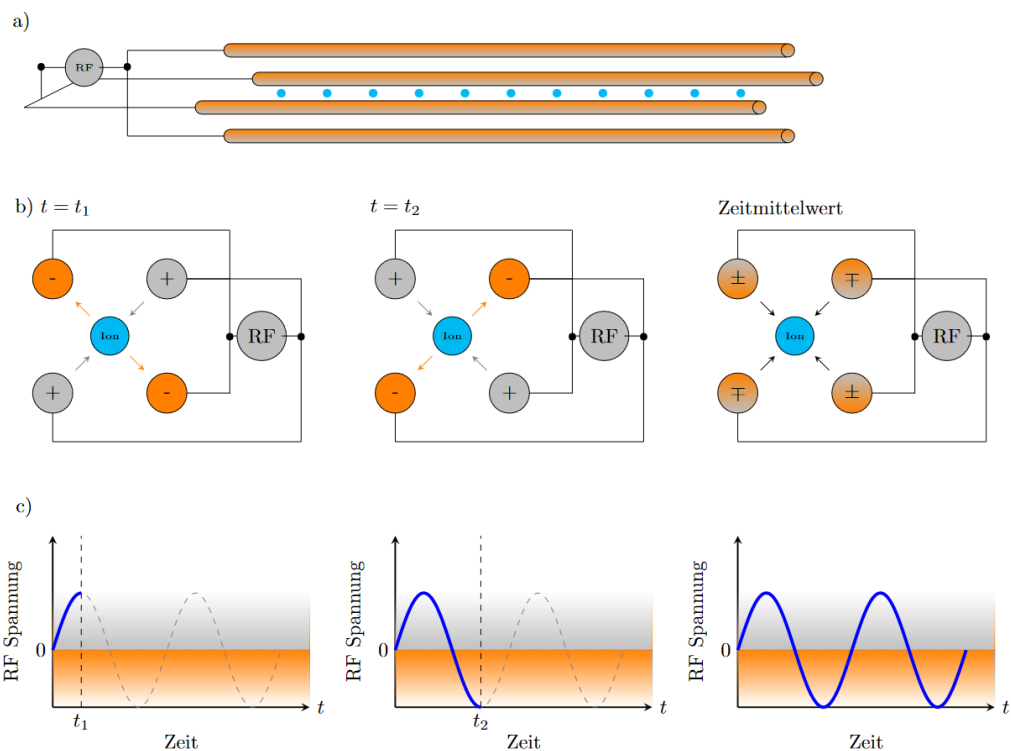


Abbildung 9: Aufbau einer linearen Paul Falle – Das Grundprinzip eines Ionen-basierten Quantencomputers. Geladene Atome oder Moleküle werden mittels oszillierender elektromagnetischer Felder eingefangen. Weitere Erklärungen sind dem Fließtext zu entnehmen. Die Graphik ist angelehnt an Referenz [H 2].

Abbildung 9 zeigt schematisch den Aufbau einer linearen Paul Falle. Diese besteht aus vier parallelen Leiterdrähten [Abbildung 9a)], die ein charakteristisches, zeitlich oszillierendes Quadrupolfeld in der zu ihnen senkrecht stehenden Ebene erzeugen. Während die gegenüberliegenden Leiter stets gleichnamig geladen sind, sind benachbarte Leiter stets entgegengesetzt geladen, was schematisch in Abbildung 9b) dargestellt ist. Weiter wird die absolute Ladung der einzelnen Drahtpaare mittels einer Wechselspannung (Radio-Frequency RF) sinus-förmig variiert [siehe Abbildung 9c)]. Eine solche Konfiguration führt dazu, dass Ionen, die sich innerhalb des Quadrupolfeldes

befinden immer weiter an das Minimum der zugehörigen Feldkonfiguration gezwungen werden. An diesem Punkt, der Symmetrieachse des Systems, heben sich die einzelnen Kräfte auf und das Ion wird »fixiert«. Der bis hierher beschriebene Aufbau der Paul-Falle kann allerdings noch nicht vollständig sein, warum? Entlang der Symmetrieachse, parallel zu den Leiterdrähten, existiert in jener Feldkonfiguration noch keine fixierende Kraft. Wie werden die Ionen also entlang der Symmetrieachse fixiert? Die Lösung ist so einfach wie genial. Das Aneinanderreihen von  $N$  gleichgeladenen Ionen mit entsprechend gleich geladenen Endelektroden führt zu einer idealerweise äquidistanten Kette an gefangenen Ionen. Grund dafür ist die abstoßende Coulomb-Wechselwirkung zwischen den einzelnen Ionen und den Endelektroden.

Unter »Quantencomputing« versteht man die kontrollierte Manipulation diskreter Energieniveaus in atomähnlichen Systemen, wie in den vorangehenden Kapiteln bereits ausgeführt. Die hier beschriebenen Ionen in der Paulfalle kann man konkret mittels hochpräziser Laserimpulse derartig manipulieren, dass mit ihnen einzelne Quantenalgorithmen realisiert werden können. Stellen wir uns der Einfachheit halber zuerst ein einzelnes gefangenes Ion in der Paulfalle vor. Mittels Lichtquanten, also kleinsten diskreten Energieportionen, ist es nun möglich die elektronischen Zustände in dessen äußerstem Orbital zu beeinflussen. In der einfachsten Apparatur eines Ionen-basierten Quantencomputers benutzt man Atome der zweiten Hauptgruppe, die ionisiert ein einzelnes Elektron im äußersten Orbital haben. Jene Systeme sind heutzutage (auch theoretisch) sehr gut verstanden. Somit erhält man durch ein derartiges Vorgehen sehr gut kontrollierbare Zwei-Niveau-Systeme. Erweitert man nun die lineare Kette um weitere Ionen, kann man die einzelnen Zwei-Niveau-Systeme (Qubits) miteinander koppeln und damit z.B. Quantenverschränkung realisieren. Zusammen definiert dies die (grobe) Struktur eines  $N$ -Qubit Ionenfallen-basierten Quantencomputers.

Autoren: Dr. Christian Tutschku (Fraunhofer IAO), Daniel Pranjic (Fraunhofer IAO)

Kontakt: [christian.tutschku@iao.fraunhofer.de](mailto:christian.tutschku@iao.fraunhofer.de), [daniel.pranjic@iao.fraunhofer.de](mailto:daniel.pranjic@iao.fraunhofer.de)

### 1.3.2 Supraleitende Architekturen

Die grundlegende Idee, der im vorigen Kapitel besprochenen Ionenfallen, ist natürlich vorkommende quantenmechanische Systeme so zu manipulieren, dass sie durch kontrollierte Operationen für Quantencomputing nutzbar gemacht werden können. Supraleitende Qubits hingegen sind elektrische Schaltkreise, die so konstruiert sind, dass sie sich trotz ihrer makroskopischen Größe quantenmechanisch verhalten. Dieser Umstand hat ihnen den informellen Namen »künstliche Atome« eingebracht [H 3], [H 4].

Die physikalischen Eigenschaften der supraleitenden Qubits ergeben sich durch die verwendeten Bauteile wie Kondensatoren, Spulen und sogenannten Josephson-Kontakten. Dadurch ist es möglich jene Qubits auf die gewünschten Anforderungen zuzuschneiden – eine Freiheit, die bei natürlich vorkommenden Quantensystemen (in dieser Art) nicht existiert. Ein Großteil der dafür benötigten Elektronikkomponenten ist standardmäßig im industriellen Maßstab verfügbar. Hierin liegt ein wichtiger Unterschied beider Technologien. Fabrizierte supraleitende Qubits sind flexibler, können aber aufgrund des Produktionsprozesses auch stark in ihren Eigenschaften variieren. Ionen-Qubits hingegen sind fundamental gleich, wenn sie aus gleichen Ionen bestehen.

Bei der Nutzung herkömmlicher Leiter anstatt Supraleiter entsteht ein kontinuierliches Spektrum. Quantencomputing basiert aber auf der Manipulation wohldefinierter, diskreter Energieniveaus (wie in Abbildung 7 zu sehen).

Um eine solche Diskretisierung zu erhalten, ist es nötig, die komplexe Bewegung der Elektronen im Leitmaterial zu ordnen. Hierfür nutzen Physiker den Effekt der Supraleitung. Dieser Effekt tritt auf, wenn man spezielle Materialien wie z.B. Aluminium oder Niobium stark kühlt. Wird eine bestimmte, kritische Temperatur unterschritten, legen die Elektronen ein Verhalten an den Tag, was (bis dato) bei alltagsüblichen Temperaturbereichen<sup>2</sup> nicht beobachtbar ist. Zum einen verschwindet der Widerstand, d.h. der Ladungstransport ist verlustfrei. Außerdem kommen die Elektronen nicht mehr einzeln, sondern nur noch zu zweit - in sogenannten Cooper-Paaren vor. Es ergibt sich ein kollektiver Zustand, der trotz mehrerer Milliarden Elektronen in gängigen Schaltkreisen mathematisch, wie ein einzelnes quantenmechanisches Teilchen beschrieben werden kann [H5, H6]. Das resultierende Energiespektrum ist nicht mehr kontinuierlich, sondern besteht nun aus diskreten Niveaus. Wie hieraus konkret die Qubit-Niveaus  $|0\rangle$  und  $|1\rangle$  definiert werden können wird im Folgenden erläutert.

---

<sup>2</sup> Auf Hochtemperatur-Supraleitung soll an dieser Stelle nicht tiefer eingegangen werden.

### 1.3.2.1 Aufbau eines supraleitenden Qubits

Um das Funktionsprinzip von supraleitenden Qubits zu verdeutlichen, betrachten wir zunächst einen sehr einfachen Schaltkreis, bestehend aus einem Kondensator und einer Spule [(Abbildung 10a)]. Dies ist ein sogenannter LC-Schwingkreis. Die in einem solchen Schaltkreis enthaltene Energie oszilliert zwischen Spule  $L$  und Kondensator  $C$  - ein Prozess, der analog zur Schwingung eines Pendels interpretiert werden kann. Um nun einen solchen Schaltkreis supraleitend zu machen, wird er in einem Kryostaten bei Temperaturen von ungefähr 20 Millikelvin betrieben. D.h. die Betriebstemperatur ist nur knapp über dem absoluten Nullpunkt, der nach dem dritten Hauptsatz der Thermodynamik niedrigsten möglichen Temperatur überhaupt. In der alltagsüblichen Skala angegeben liegt der absolute Nullpunkt bei  $-273,15$  Grad Celsius. Zum Vergleich: Die Durchschnittstemperatur im Weltraum beträgt ca. 3 Kelvin bzw.  $-270,15$  Grad Celsius. Durch die im vorigen Abschnitt beschriebenen Effekte werden die Energieniveaus hierbei diskretisiert – der Schwingkreis kann nur noch bei bestimmten Frequenzen schwingen.

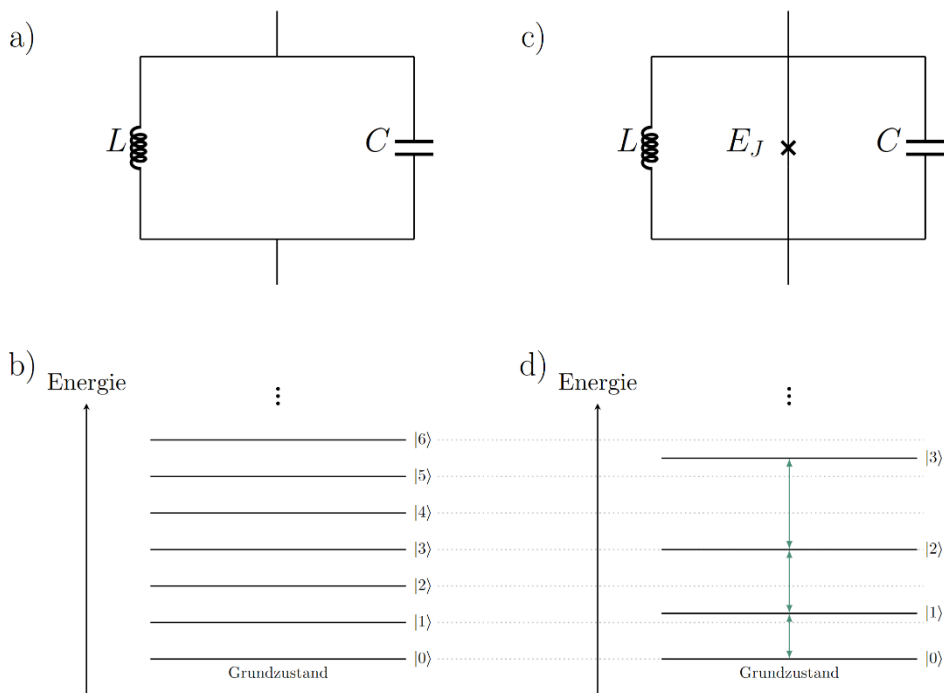


Abbildung 10: a) Elektronischer Schaltplan eines LC-Schwingkreises; b) zugehöriges quantisiertes Energiespektrum (sogenannter harmonischer Oszillator); c) einfaches Qubit, realisiert durch eine parallel geschaltene Induktivität  $L$ , einen Josephson-Kontakt mit Energie  $E_J$  und eine Kapazität  $C$ ; d) zugehöriges Energiespektrum des Transmon-Qubits. Der bedeutende Unterschied zwischen dem Energiespektrum des supraleitenden LC-Schwingkreises in (b) und dem des Transmon-Qubits in (d) liegt in der Anharmonizität. Während in (b) die diskreten Energieniveaus äquidistant zueinander liegen, vergrößert sich diese Differenz mit zunehmender Energie beim Transmon-Qubit. Um den Unterraum aller Zustände auf das  $|0\rangle$  und  $|1\rangle$  Subsystem zu beschränken, muss diese Anharmonizität erzeugt werden. Anderenfalls würde eine Anregung von  $|0\rangle$  nach  $|1\rangle$  gleichzeitig auch  $|1\rangle$  nach  $|2\rangle$ ,  $|2\rangle$  nach  $|3\rangle$  usw. induzieren.

Das entstehende Spektrum ist in Abbildung 10b) zu sehen und zeigt, dass die Energiedifferenz zwischen den einzelnen diskreten Energieniveaus in solch einem System jeweils gleich groß ist. Für die Verwendung als Qubit ist dies problematisch, da bei der Anregung eines Übergangs gleichzeitig alle anderen Übergänge mit angeregt werden und man somit kein konkretes Zwei-Niveau-System isolieren kann. Das zentrale Bauteil, das aus einem LC-Schaltkreis ein Qubit macht, ist der sogenannte Josephson-Kontakt (Abbildung 11). Hier wird in ein Stück supraleitendes Material eine Barriere eingefügt. Die Barriere besteht aus nichtleitendem Material, sodass der reguläre Stromfluss unterbrochen ist. Ist die Barriere dünn genug (wenige Nanometer), können einzelne Cooper-Paare sie dennoch überwinden. Dies ist ein quantenmechanischer Effekt und wird als »Tunneln« bezeichnet<sup>3</sup>. Eingebaut in einen Schaltkreis [(Abbildung 10c)] agiert der Josephson-Kontakt wie eine nichtlineare Spule [H 7]. Die mathematischen Details würden den Rahmen dieser Studie sprengen, weshalb wir hier den interessierten Leser auf Referenz [H 8] verweisen. Wichtig ist an dieser Stelle jedoch der Effekt, welchen der Josephson-Kontakt auf das Energiespektrum hat. Dieser ist in Abbildung 10d) zu sehen. Anders als beim reinen LC-Schwingkreis sind die Abstände der Energieniveaus in einem solchen Schaltkreis nicht mehr konstant. Dadurch kann man näherungsweise die unteren beiden Niveaus  $|0\rangle$  und  $|1\rangle$  als »isoliertes« Zwei-Niveau-System interpretieren und somit als Qubit nutzen (vgl. Abbildung 7).

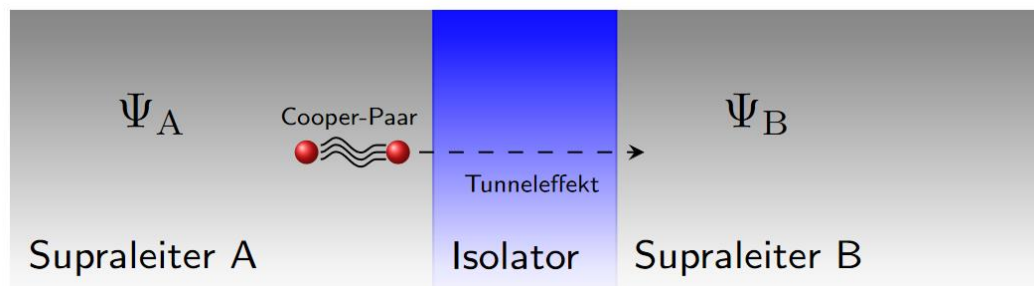


Abbildung 11: Schematische Veranschaulichung der Funktionsweise eines Josephson-Kontakts. Eine dünne (wenige Nanometer breite) Isolator-Barriere aus nichtleitendem Material wird in einen Supraleiter eingefügt, wodurch zwei Supraleiter entstehen. Deren Cooper-Paare, die für die Supraleitung verantwortlich sind, können jeweils durch die Wellenfunktionen  $\Psi_{A/B}$  beschrieben werden. Ist der Isolator hinreichend dünn kommt es zum quantenmechanischen Phänomen des »Tunnelns«, wobei sich die jeweiligen Wellenfunktionen in die Bereiche des anderen Supraleiters ausdehnen können. Es resultiert eine nicht-lineare Strom-Spannungscharakteristik, die für das Transmon-Qubit verwendet wird.

Alle gängigen supraleitenden Schaltkreise teilen dieses Grundprinzip. Die genauen Eigenschaften der verwendeten Bauteile wie Kondensatoren und Spulen sowie die

<sup>3</sup> Für die theoretische Beschreibung des Tunnelns von Cooper-Paaren durch eine Barriere hat Brian D. Josephson im Jahr 1973 den Nobelpreis erhalten.



Anordnung und Anzahl an Josephson-Kontakten erlauben eine enorme Designfreiheit in der Konstruktion der Qubits. Dies hat in den letzten Jahren zu einer großen Vielfalt an verschiedenen Qubit-Varianten, wie z.B. Transmon [H 3], Xmon [H 9], Fluxonium [H 10] oder Unimon [H 11] geführt. Verschiedene Qubit Typen haben jeweils individuelle Vor- und Nachteile und unterscheiden sich u.a. in der Kohärenzzeit und den realisierbaren Gates.

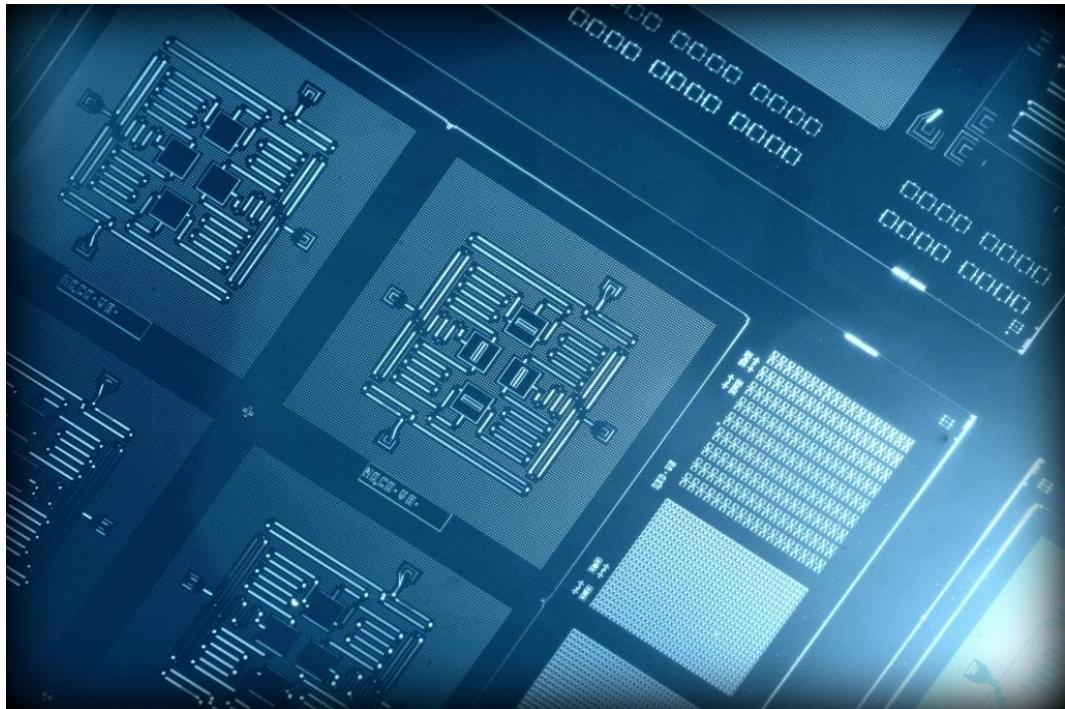


Abbildung 12: Illustration eines supraleitenden Quantenchips mit vier Transmon Qubits. Zu erkennen sind die Resonatoren, die an die Transmons gekoppelt sind, sowie die Mikrowellenleiter, mit denen die Transmons über passende Pulssequenzen initialisiert, kontrolliert und ausgelesen werden ©IBM.

### 1.3.2.2 Kontrolle und Operation

Die Zustände von supraleitenden Qubits können über klassische Kontrollelektronik gesteuert werden. Hierfür koppelt man den Schaltkreis an eine externe »Transmissionline«. Um Übergänge zwischen den Zuständen z.B. von  $|0\rangle$  nach  $|1\rangle$  anzuregen, speist man elektromagnetische RF-Signale ein. Die Frequenz der Signale muss dem Energieunterschied zwischen den Energieniveaus entsprechen. Diese ist üblicherweise im niedrigen Gigahertz Bereich und entspricht somit Mikrowellenstrahlung.

Wie in Abbildung 12 zu sehen ist, können mehrere supraleitende Qubits auf einen gemeinsamen Chip aufgetragen werden, um ein Multi-Qubit-System zu erzeugen. Die einzelnen Qubits können beispielsweise kapazitiv über Kondensatoren oder induktiv über Spulen gekoppelt werden. Um die Qubits miteinander wechselwirken zu lassen, werden externe Kontrollpulse verwendet. Die Intensität und Dauer der Wechselwirkung ist in der

Regel über die Gestaltung der Kontrollpulse steuerbar. Ein prominentes Beispiel für die Realisierung eines Zwei-Qubit-Gates ist das Cross Resonance Gate

[H 12], bei dem Qubit 1 mit der Resonanzfrequenz des benachbarten Qubits 2 angeregt wird. Mit ihm und weiterer Einzel-Qubit-Gates lässt sich z. B. das CNOT Gate realisieren, welches im Quantencomputing ein zentraler Bestandteil der Algorithmen ist. Grund dafür ist, dass mit jenem Gate maximal verschränkte Qubit-Paare erzeugt werden können. Dabei ist die optimale Form der Mikrowellenpulse entscheidend, denn um die Fidelität – das Maß für die Ähnlichkeit des erzeugten mit dem gewünschten Zustand – zu maximieren, werden unter anderem Echo-Pulse und Obertöne verwendet. Auf diese Weise werden ungewünschte Störeffekte unterdrückt [H 13]. Das Cross Resonance Gate ist nicht das einzige Zwei-Qubit-Gate mit dem das CNOT Gate realisiert werden kann, z. B. ist dies auch mit dem iSWAP möglich [H 2]. Abschließend sei angemerkt, dass die Art der realisierbaren Gates stark Hardware-abhängig ist und sich stark zwischen den verschiedenen Chiparchitekturen unterscheidet.

Supraleitende Schaltkreise müssen aufwendig gekühlt werden, damit die Chips einen quantenmechanischen Charakter aufweisen. Im Folgenden wird mit den diamantbasierten NV-Zentren eine Plattform vorgestellt, die auch bei Raumtemperatur betrieben werden kann.

Autoren: Dr. Marco Roth (Fraunhofer IPA), Dr. Christian Tutschku (Fraunhofer IAO), Daniel Pranjic (Fraunhofer IAO)

Kontakt: [marco.roth@ipa.fraunhofer.de](mailto:marco.roth@ipa.fraunhofer.de), [christian.tutschku@iao.fraunhofer.de](mailto:christian.tutschku@iao.fraunhofer.de), [daniel.pranjic@iao.fraunhofer.de](mailto:daniel.pranjic@iao.fraunhofer.de)

### 1.3.3 Stickstoffvakanz-Zentren in Diamant

Diamanten bestehen aus einer gitterförmigen Aneinanderreihung von Kohlenstoffatomen, wie in Abbildung 13 (links) dargestellt. Während reine Diamanten aufgrund ihrer Härte hauptsächlich in der Schmuck-, bzw. der produzierenden Industrie verwendet werden, haben Diamanten mit Verunreinigungen und Fehlstellen einzigartige Eigenschaften, die sie für die Wissenschaft – insbesondere für Quantensensing und Quantencomputing – interessant machen. Die einfachste Fehlstelle im Diamanten entsteht durch das Fehlen eines Kohlenstoffatoms in der sonst regelmäßigen Struktur. Dieser Defekt wird als Vakanz bezeichnet und ist in Abbildung 13 rechts mit einem V dargestellt. Kohlenstoffatome können aber nicht nur fehlen, sie können auch durch andere Elemente wie z.B. Stickstoff (chemisches Symbol N) ersetzt werden. Liegen eine Vakanz und ein

Stickstoffatom benachbart vor, wie in Abbildung 13 (rechts) gezeigt, entsteht ein Stickstoffvakanz-Zentrum (NV-Zentrum [H 14], [H 15]).

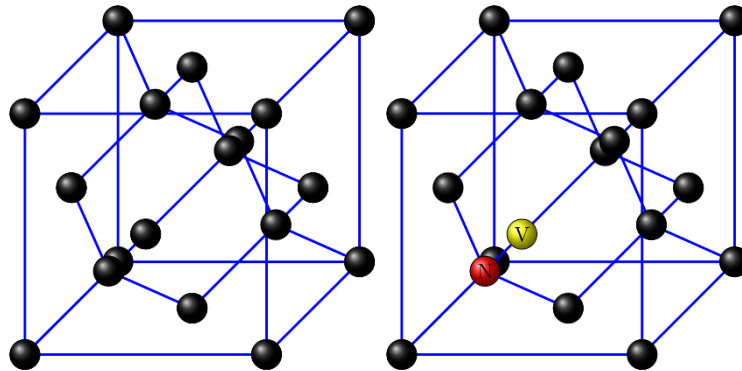


Abbildung 13: Reines Diamantkristallgitter (links) und Präsenz eines Stickstoffvakanz- (NV-) Zentrums (rechts) in der ansonsten geordneten Struktur der Kohlenstoffatome (schwarze Kugeln). Weitere Erklärungen sind dem Fließtext zu entnehmen.

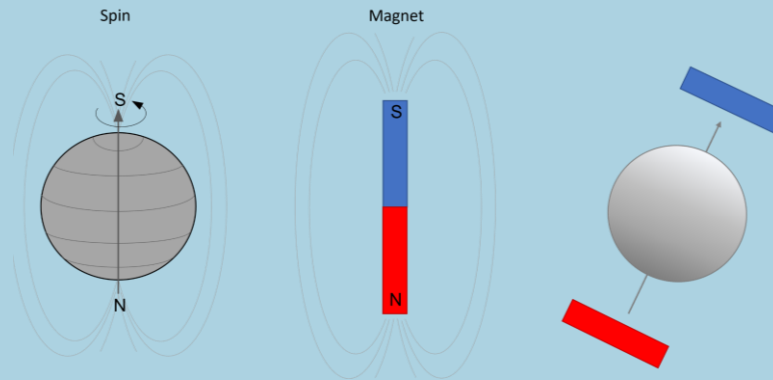
### 1.3.3.1 Aufbau eines Stickstoffvakanz-Zentrums

Das Stickstoffatom und die benachbarte Fehlstelle bilden eine Einheit, deren Energiekonfiguration als Qubit nutzbar ist. Um die nötige Zweiniveaustruktur zu erhalten, nutzt man maßgeblich die quantenmechanischen Eigenschaften des zugehörigen Elektronspins (siehe Infobox 9: **Spin**).

Ein Elektron kann in zwei Spinzuständen (»spin-up« und »spin-down«) vorliegen. Ohne äußere Einflüsse sind die beiden Zustände entartet, d.h. sie haben die gleiche Energie. Legt man nun ein externes Magnetfeld an, reduziert sich die Energie des einen Spinzustands während sich die Energie des anderen erhöht. Die so entstandenen Zustände mit unterschiedlichen Energieniveaus bilden ein Zwei-Niveau-System und können als Qubit genutzt werden.

Das auf diese Art entstandene Qubit hat einige bemerkenswerte Eigenschaften. So zeichnet es sich bei Raumtemperatur durch eine lange Kohärenzzeit aus. Diese liegt für das beschriebene Elektronenspin-Qubit bei 2,4 Millisekunden [H 16] ( $T_2$  bei Raumtemperatur und mehrere Sekunden bei tieferen Temperaturen [H 17]). Wie im nächsten Kapitel 1.3.3.2 im Detail beschrieben, können zum NV-basierten Quantencomputing insbesondere Kernspin-Qubits genutzt werden. Ein Einzel-Qubit Zustand kann über eine Minute erhalten werden [H 18].

### Infobox 8: Spin



Der Spin (englisch Drehung) ist eine quantenmechanische Eigenschaft von Teilchen. Mathematisch besitzt er dieselben Eigenschaften wie der klassische Drehimpuls mit zwei maßgeblichen Unterschieden: 1) Der Spin wird nicht durch eine Drehbewegung hervorgerufen, sondern ist eine intrinsische Eigenschaft eines quantenmechanischen Teilchens. 2) Anders als der klassische Drehimpuls kann der Spin nur in halb- oder ganzzahligen Vielfachen des planckschen Wirkungsquantums auftreten.

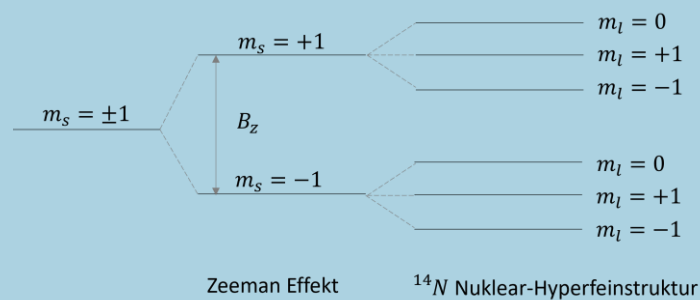
Unter diesen Einschränkungen wirken Spin-behaftete Teilchen wie kleine Kreisel oder Gyroskope, deren Spin die Drehrichtung bestimmt. Gleichzeitig ist der Spin eng verknüpft mit den magnetischen Eigenschaften von Teilchen. Geladene Teilchen mit Spin verhalten sich wie kleine Magnete, bei denen die Magnetisierungsachse durch den Spin gegeben ist. Sie können sich in einem externen Magnetfeld drehen (wie rechts in der Abbildung gezeigt), anziehen oder abstoßen – wie echte Magnete.

#### 1.3.3.2 . Multi-Qubit-Systeme in Diamant

Während die aufgespalteten Energiezustände des Elektronenspins im NV-Zentrum ein Zwei-Niveau-System bilden, können im Wesentlichen – wie bereits angedeutet – Qubits mit sehr langen Kohärenzzeiten realisiert werden, indem Kernspinzustände genutzt werden. Das NV-Zentrum ist, wie in Abbildung 13 gezeigt, von Kohlenstoffatomen umgeben. Es gibt zwei stabile Kohlenstoff-Isotope, die sich in der Anzahl an Neutronen unterscheiden. Das mit einer Häufigkeit von 98,89% am weitesten verbreitete Isotop ist das  $^{12}\text{C}$ . Es besteht aus 6 Protonen und 6 Neutronen. Das mit 1,11% seltener vorkommende Isotop  $^{13}\text{C}$  hat ein zusätzliches Neutron bei der gleichen Anzahl an Protonen. Während  $^{12}\text{C}$  einen neutralen Kernspin hat und sich somit nicht als Qubit eignet, hat  $^{13}\text{C}$  einen Kernspin von  $1/2$  und kann als Qubit verwendet werden. Ähnlich wie bei der

bereits beschriebenen Aufspaltung der Elektronenspins durch den Zeemann-Effekt, können Kernspinzustände zwei diskrete Energieniveaus erzeugen. Die Aufspaltung der Kernspinniveaus benötigt aber anders als beim Elektronenspin kein externes Magnetfeld, sondern entsteht auf natürliche Weise durch die Interaktion des NV-Zentrums mit seiner Umgebung. Die so entstehende Aufspaltung wird Hyperfeinstruktur, die dafür verantwortliche Kopplung Hyperfeinkopplung genannt (siehe Infobox 10).

#### Infobox 9: Hyperfeinstruktur



In der Infobox 9 wurden bereits die Wechselwirkung von Elektronenspins mit äußeren Magnetfeldern und die damit verbundene Aufspaltung der Energieniveaus beschrieben. Analog zu diesem Effekt bezeichnet die Hyperfeinstruktur die Aufspaltung von Kernspinzuständen in verschiedene Energieniveaus. Die Energiedifferenz der daraus entstehenden Zustände ist deutlich kleiner als die der Elektronenspins, woraus sich der Name Hyperfeinstruktur ableitet. Das zur Aufspaltung nötige Magnetfeld muss in diesem Fall nicht von außen angelegt werden, sondern wird natürlicherweise durch die Spins der benachbarten Teilchen erzeugt (siehe Infobox 8: **Spin**).

Der Diamant-Quantenprozessor besteht aus einem NV-Zentrum mit einer lokalen Ansammlung von hyperfein-gekoppelten Kernspins (Abbildung 14). Die Zwei-Niveau-Systeme, die durch die Kernspins ( $^{13}\text{C}$  und  $\text{N}$ ) entstehen, werden als Qubits genutzt. Der Elektronenspin des NV-Zentrums hingegen dient als Quantenbus. Dieser initialisiert die gekoppelten Kernspins, vermittelt Wechselwirkungen zwischen ihnen und erlaubt das Auslesen der Zustände.

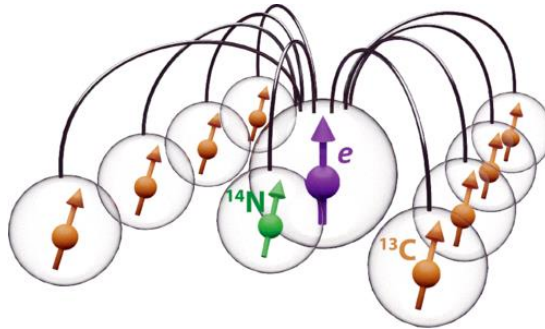


Abbildung 14: Illustration eines NV-Zentrums in einer lokalen Ansammlung von hyperfein-gekoppelten  $^{13}\text{C}$  Kernspins<sup>4</sup>. Der Elektronenspin koppelt an die Nucleons ( $^{13}\text{C}$  und  $\text{N}$ ), die wiederum als Qubits genutzt werden. Die Kernspins besitzen aufgrund der lokalen Felder im Kristall ihre jeweils eigene Resonanzfrequenz. Üblicherweise würden die Kernspins den Elektronenspin stören, jedoch kann über Spin-Echos, der Spin bis zu einer Größenordnung von 1 Sekunde geschützt werden [H19].

Die Hyperfeinkopplung zwischen dem NV-Elektronenspin und den Kernspins führt zu einer Aufspaltung der elektronischen und nuklearen Energieniveaus. Diese Aufspaltung hängt von der jeweiligen Wechselwirkungsstärke der Hyperfeinkopplung zwischen jedem Kern- und dem NV-Elektronenspin, sowie von den jeweiligen Elektronen- und Kernspinzuständen ab. Um ein Register zu bilden, ist es nötig, Kernspins mit unterschiedlicher Hyperfeinkopplungsstärke zu wählen, um jedes Kernspin-Qubit im Register einzeln adressieren zu können. Das so entstandene Register zeichnet sich durch lange Kohärenzzeiten aus und kann als Quantenspeicher genutzt werden

Anschließend können mehrere Register beispielsweise mittels Spin-Photonen-Verschränkung gekoppelt werden. Dies ist in Abbildung 15 schematisch dargestellt. Dabei sendet ein Register ein Photon (siehe Kapitel 1.3.4) aus, welches mit dem Spin-Zustand des anderen Zentrums verschränkt wird. Durch das geschickte Zusammenführen der Photonen auf einen Strahlteiler können weit entfernte NV-Zentren so miteinander verschränkt werden.

<sup>4</sup> Grafik übernommen von: Bradley, C.E. et al., A Ten-Qubit Solid State Spin Register with Quantum Memory up to One Minute, Link: <https://journals.aps.org/prx/abstract/10.1103/PhysRevX.9.031045>, unter der Creative Commons 4.0 Lizenz: <http://creativecommons.org/licenses/by/4.0/>.

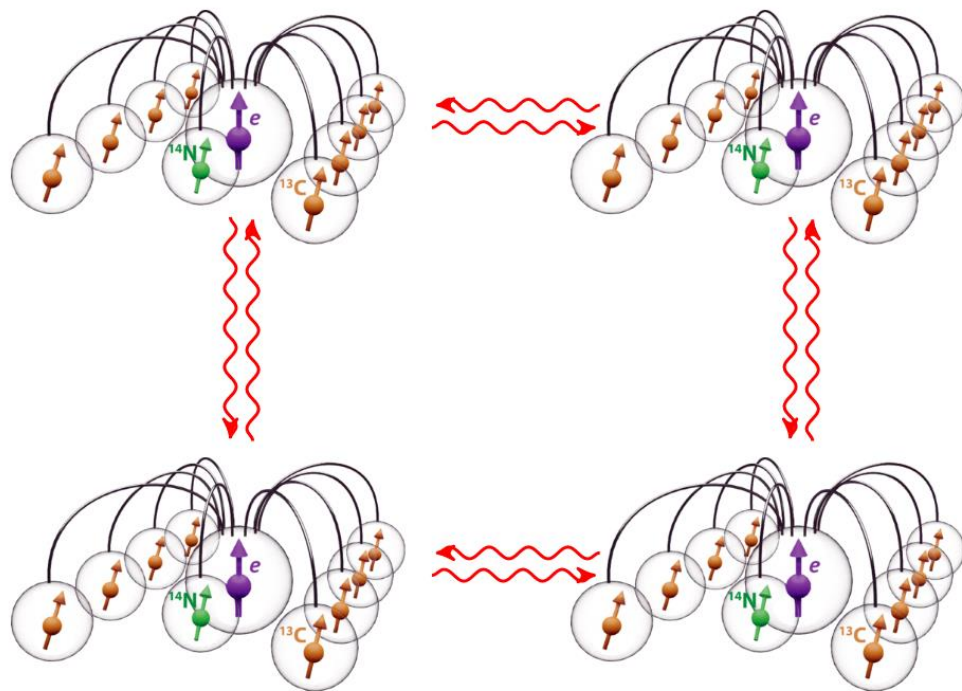


Abbildung 15: Schematische Darstellung für die Skalierung von NV-Registern. Detaillierte Informationen sind dem Fließtext zu entnehmen<sup>5</sup>.

Die Nutzung von NV-Zentren für Quantencomputing birgt mehrere Vorteile. So haben die genutzten Qubits in der Regel eine lange Kohärenzzeit und einen relativ einfachen Skalierungsmechanismus. Weiterhin ist der Betrieb bei Raumtemperatur möglich, was grundsätzlich einen dezentralen Einsatz als Edge-Gerät erlaubt. Diese Eigenschaften können beispielsweise ebenfalls dazu genutzt werden einen Quantenspeicher für Berechnungen auf anderen Quantencomputerarchitekturen bereitzustellen.

Autoren: Kathrin König (Fraunhofer IAF), Dr. Farida Shagieva (Universität Stuttgart HLRS), Daniel Pranjic (Fraunhofer IAO), Dr. Christian Tutschku (Fraunhofer IAO)

Kontakt: [kathrin.koenig@iaf.fraunhofer.de](mailto:kathrin.koenig@iaf.fraunhofer.de), [daniel.pranjic@iao.fraunhofer.de](mailto:daniel.pranjic@iao.fraunhofer.de), [christian.tutschku@iao.fraunhofer.de](mailto:christian.tutschku@iao.fraunhofer.de)

<sup>5</sup> Grafik übernommen bzw. weiterentwickelt/gekoppelt: Bradley, C.E. et al., A Ten-Qubit Solid State Spin Register with Quantum Memory up to One Minute, Link: <https://journals.aps.org/prx/abstract/10.1103/PhysRevX.9.031045>, unter der Creative Commons 4.0 Lizenz: <http://creativecommons.org/licenses/by/4.0/>.

### 1.3.4 Photonen

Elektromagnetische Strahlung ist allgegenwärtig. Die bekannteste Form ist das für den Menschen sichtbare Licht. Jenseits des sichtbaren Bereichs tritt diese Strahlung z.B. in Form von Infrarot- oder Röntgenstrahlung in Erscheinung. Elektromagnetische Strahlung weist sowohl Eigenschaften von Teilchen als auch von Wellen auf. Dies wird als Welle-Teilchen-Dualismus bezeichnet. In der Teilchenbetrachtungsweise besteht elektromagnetische Strahlung aus Photonen. Unter kontrollierten Bedingungen können diese als Qubits verwendet werden [H 20].

Ein wesentlicher Vorteil von photonischen Qubits besteht darin, dass sie ihre Quantenzustände für sehr lange Zeit beibehalten. Sie wechselwirken in erster Näherung nicht direkt miteinander und können daher nur absorbiert oder gestreut werden. So sind Photonen von fernen Sternen und Galaxien zum Teil mehrere Milliarden Jahre unverändert unterwegs, bevor sie unsere Augen erreichen. Die Photonen des Lyman-Alpha-Klecks etwa können immer noch in ihrem ursprünglichen Zustand polarisiert sein, wenn sie nach einer Reise von 11,5 Milliarden Jahren die Erde erreichen. Hinsichtlich des Quantencomputings impliziert jene Eigenschaft allerdings ebenfalls, dass Zwei-Qubit-Operationen für photonische Systeme nicht-trivial zu konstruieren sind [H 15]. Die Funktionsweise der ersten beiden Methoden wird im Folgenden beschrieben. Beim Boson Sampling handelt es sich um ein kompliziertes Verfahren, was kein universelles Quantencomputing erlaubt. Es wird daher in dieser Studie nicht weiter behandelt.

#### **Gatterbasiertes Modell**

Im Gattermodell wird ein Qubit durch die horizontale ( $|H\rangle$ ) und vertikale ( $|V\rangle$ ) Polarisation eines einzelnen Photons kodiert (siehe Abbildung 16, Infobox 11). Um die Polarisation eines Photons zu ändern, d.h. Gates zu realisieren, werden sogenannte Wellenplatten verwendet. Durch die Kombination mehrerer Wellenplatten, kann eine beliebige Drehung der Polarisation und daher des Qubit-Zustands erzeugt werden (siehe Poincaré-Kugel in Abbildung 16 rechts).

Im gatterbasierten Quantencomputing Modell mittels Photonen werden meist Einzel-Photonenquellen verwendet [H 22], [H 23] und das Experiment bzw. der Algorithmus viele Male wiederholt, um eine ausreichende Anzahl an Photonen auf dem Detektor zu akkumulieren.



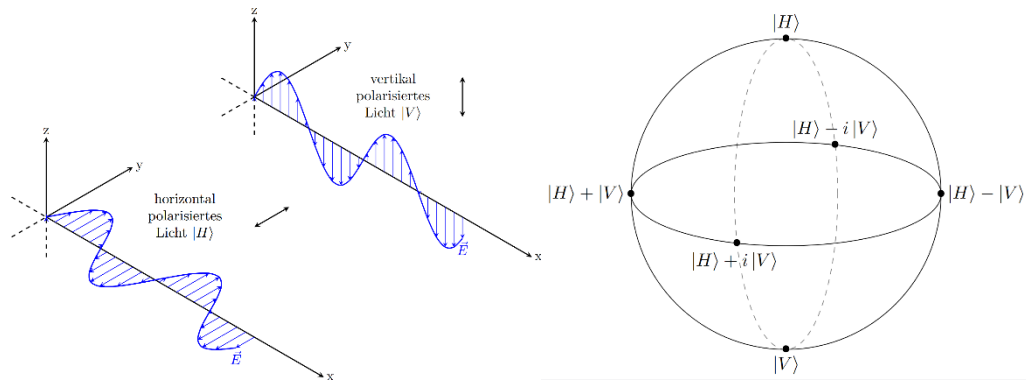
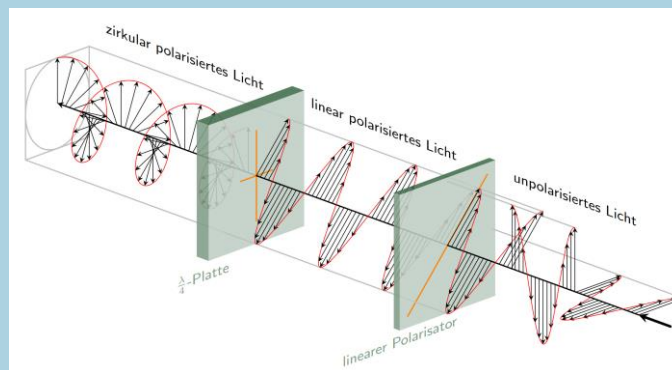


Abbildung 16: (Links) Die Basiszustände eines photonischen Qubits ( $|0\rangle$  und  $|1\rangle$ ) können über die Polarisationszustände der Lichtwelle  $|H\rangle$  (horizontal) und  $|V\rangle$  (vertikal) kodiert werden. Analog zur Bloch-Kugel für ein Qubit existiert für die Polarisationszustände der Lichtwelle die sogenannte Poincaré-Kugel (rechts). Um eine Superposition der horizontalen und vertikalen Polarisation zu erzeugen, kann eine sogenannte Lambda-halbe Wellenplatte genutzt werden.

### Infobox 10: Polarisation



Lichtwellen stellen elektromagnetische Wellen dar, welche sich durch sich gegenseitig induzierende, zeitlich oszillierende elektrische- und magnetisch Felder ausbreiten. Jene Felder schwingen in der Ebene senkrecht zur Ausbreitungsrichtung. Diese Ebene wird als Polarisations Ebene bezeichnet. In jener Ebene können elektrische und magnetische Polarisationsvektoren zeitlich in konstante Richtung zeigen, dann redet man von linear polarisiertem Licht, oder sich zeitlich gleichmäßig um die Ausbreitungsrichtung drehen. In diesem Fall redet man von zirkular polarisiertem Licht. Für viele physikalische Experimente werden Lichtquellen benötigt, die eine strenge lineare Polarisation besitzen. Dies ist zum Beispiel bei vielen Lasern der Fall. Übrigens ist auch das Licht von Smartphones oft linear polarisiert. Das ist leicht zu erkennen, wenn man durch eine Sonnenbrille mit Polarisationsfilter auf den Bildschirm schaut: Wenn man das Handy dreht, wird der Bildschirm ab einem bestimmten Moment schwarz. Dies bedeutet, der Polarisationsvektor des Lichtes und die stark absorbierende Komponente des Polarisationsfilters stehen zueinander parallel.

## Messungsbasiertes Modell

Universelle Quantenschaltkreise mittels photonischer Systeme können auch durch Qubit-Messungen realisiert werden. Dieses Konzept bezeichnet man als messungsbasiertes Quantencomputing [H 24]. Hierbei verwendet man einen hohen Grad an Verschränkung als initiale Ressource und anschließend lokale Messungen auf einzelnen Qubits um Berechnungen (Algorithmen) durchzuführen

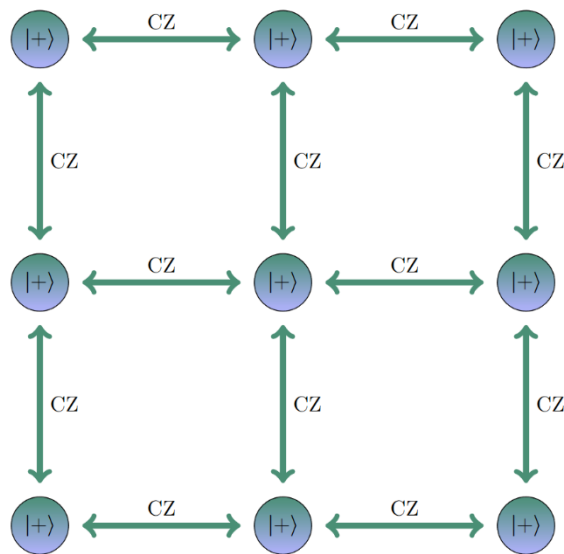


Abbildung 17: Clusterzustände sind eine spezielle Art von Ressourcenzustand, die im messungsbasierten Quantencomputing Verwendung finden. Die einzelnen Qubits bilden die Eckpunkte „vertices“ eines Graphen. Die Qubits werden mit controlled-Z (CZ) Gattern untereinander verschränkt. Mithilfe von lokalen Messungen können anschließend beliebige unitäre Gatter realisiert werden.

Die Basis für derartige Berechnungen ist ein sogenannter Ressourcenzustand (Abbildung 17). Unter einem Ressourcenzustand versteht man eine Menge von Qubits, die die Knoten eines Graphen darstellen. Die Kanten entsprechen den Interaktionen zwischen diesen Qubits. Universelle Quantenberechnungen im messungsbasierten Modell sind anschließend genau dann möglich, wenn man Einzel-Qubit-Rotationen und Zwei-Qubit Verschränkungsoperationen (wie z.B. CNOT oder Controlled-Z) realisieren kann [H 25].

Da photonische Systeme wie erwähnt kaum miteinander wechselwirken jedoch miteinander sehr gut verschränkt werden können (effiziente Präparation von Ressourcenzuständen), eignen sich jene Systeme sehr gut für messungsbasierte Quantenberechnungen. Die Weiterentwicklung jenes Ansatzes bezeichnet man als fusionsbasiertes Quantencomputing, welches im Rahmen dieser Studie jedoch nicht vertieft werden soll [H 26].

Hinsichtlich der Skalierung von optischen Quantenschaltungen verwendet man beispielsweise integrierte Wellenleiter auf Nanochips (Abbildung 18). Diese lassen sich mittels heutiger Expertise in der Nanotechnologie miniaturisieren. Photonische Quantenschaltungen mit integrierter Nanophotonik sind programmierbare Chips, die bei Raumtemperatur arbeiten und mit einem vollautomatischen Kontrollsystem verbunden sind (Abbildung 18). Im Gegensatz zu früheren photonischen Geräten verfügen heutige photonische Quantenprozessoren über nichtklassische Lichtquellen, wie zum Beispiel sogenanntes gequetschtes Licht (siehe Infobox 12: **Gequetschtes Licht**). Auf photonischen Nanochips nutzt man außerdem pfadkodierte Bell-Zustände, um Berechnungen bzw. Algorithmen durchzuführen. Diese sind der Vollständigkeit halber in Abbildung 19 beschrieben.

#### Infobox 11: **Gequetschtes Licht**

Gequetschtes Licht ist eine Art des nichtklassischen Lichtes. Analog zur Heisenbergschen Unschärferelation für Ort und Impuls eines quantenmechanischen Teilchens existiert ebenfalls eine Unschärferelation für Amplitude und Phase einer Lichtwelle. Für kohärentes Licht ist die Unschärfe des elektrischen Feldes unabhängig von der Phase. Bei gequetschtem Licht hängt die Unschärfe des elektrischen Feldes jedoch von der Phase ab. Dies führt dazu, dass für manche Phasenbereiche die Amplitudenunschärfe beispielsweise *gequetscht* während die Phasenunschärfe vergrößert ist.

Gequetschtes Licht wird in vielen Bereichen eingesetzt, da es Messungen mit reduziertem Quantenrauschen ermöglicht. Ein bekanntes Beispiel für dessen Einsatz ist die Messung der von Einstein vorhergesagten Gravitationswellen mittels LIGO und Virgo [H 27].

In Kombination mit den sogenannten ‚approximativen‘ Gottesman-Kitaev-Preskill (GKP) Zuständen wird gequetschtes Licht insbesondere in Xanadu’s photonischen Quantencomputern eingesetzt [H 28]. Für die Erzeugung der GKP Zustände wird Gauß’sches Boson Sampling benutzt.

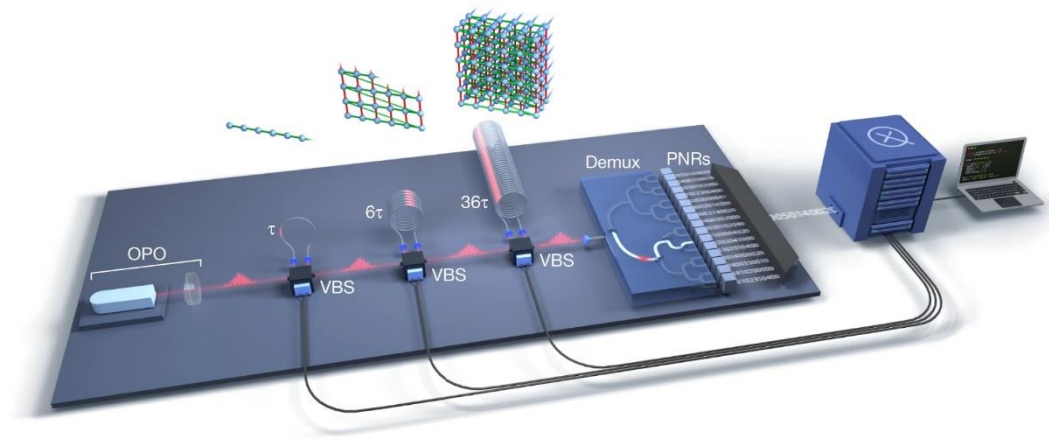


Abbildung 18: Schematische Darstellung eines photonischen Quantenprozessors der Firma Xanadu. Für die detaillierte Beschreibung der Funktionsweise des Prozessors sei auf Referenz [H 29] verwiesen aus der die Grafik entnommen wurde. Man präpariert einen Quantenzustand über Boson Sampling, welcher dann in der QPU verarbeitet wird.

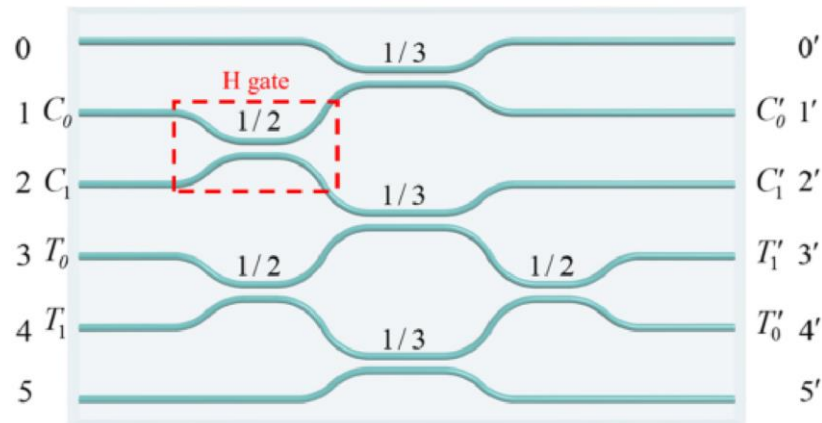


Abbildung 19 Schematische Darstellung eines photonischen Quantenchips, der aus einem H- und einem CNOT-Gatter besteht, um pfadkodierte Bell-Zustände zu erzeugen. Der Richtkoppler im rot gestrichelten Kasten stellt ein Hadamard-Gatter dar, der Rest des Chips ein CNOT-Gatter. Die mit  $C_0$  und  $C_1$  markierten Wellenleiter stellen das Control-Qubit dar, während  $T_0$  und  $T_1$  für das Target-Qubit stehen. Die übrigen Wellenleiter sind Hilfs-Qubits (Ancillas), welche für die Umsetzung der Gatteroperationen auf diesem Chip benötigt werden. Jene Grafik wurde aus Referenz [H 30] übernommen.

Autoren: Daniel Pranjic (Fraunhofer IAO), Dr. Christian Tutschku (Fraunhofer IAO),  
Dr. Farida Shagieva (Universität Stuttgart HLRS)

Kontakt: [daniel.pranjic@iao.fraunhofer.de](mailto:daniel.pranjic@iao.fraunhofer.de), [christian.tutschku@iao.fraunhofer.de](mailto:christian.tutschku@iao.fraunhofer.de)

## 1.4 Quantenalgorithmen

In diesem Abschnitt stellen wir einige der wichtigsten Quantenalgorithmen vor. Wir beginnen mit einem kurzen Abschnitt zu den historisch ersten Quantenalgorithmen, werden diese aber nicht ins Detail vertiefen, da sie nur von theoretischem Interesse sind. Anschließend stellen wir mit der Quanten-Fouriertransformation eine der bedeutendsten Subroutinen im Quantencomputing vor und erläutern mögliche Beschleunigungen. Bereits in Kapitel 1.1 wurde thematisiert, dass heutige Quantencomputer noch von geringer Größe und fehlerbehaftet sind. Algorithmen für fehlerkorrigierte Quantencomputer, die in Abschnitt 1.4.3 vorgestellt werden, sind also insbesondere für zukünftige Quantencomputer relevant. Für den Einsatz auf heutigen Quantencomputern konzipiert sind variationelle Quantenalgorithmen, welche wir in Abschnitt 1.4.4 besprechen.

### 1.4.1 Frühe Quantenalgorithmen

Im Jahr 1985 entwickelte David Deutsch den ersten Quantenalgorithmus [A 1] der demonstrieren konnte, dass ein Quantencomputer einen klassischen Computer übertreffen kann. Als Generalisierung dieses Algorithmus wurde 1992 der Deutsch-Jozsa-Algorithmus [A 2] präsentiert. Er stellt eines der ersten Beispiele für einen exponentiellen Speed-Up eines Quantenalgorithmus gegenüber jedem möglichen deterministischen, klassischen Algorithmus dar. Obwohl aus theoretischer Sicht sehr relevant, sind beide Algorithmen jedoch von geringem praktischem Nutzen, sodass wir in dieser Studie nicht weiter auf diese eingehen.

Quellen: [A 3, Kapitel 1.4], [A 4, Kapitel 7.3 und 7.5]

### 1.4.2 Quanten-Fouriertransformation

Die diskrete Fourier-Transformation (DFT, Discrete Fourier Transform) ist eine der wichtigsten mathematischen Transformationen und findet Anwendung u. a. in der digitalen Signal- und Bildverarbeitung. Ebenso spielt sie bei der Lösung von partiellen Differentialgleichungen oder der effizienten Implementierung anderer Algorithmen (z. B. zur Berechnung von Faltungen) eine herausragende Rolle. Mit der schnellen Fourier-Transformation (FFT, Fast Fourier Transform) ist ein effizienter Algorithmus bekannt, um die DFT zu berechnen. Für eine Sequenz von  $N$  Daten kann dieser die DFT mit der Komplexität  $\mathcal{O}(N \log N)$  berechnen (zur  $\mathcal{O}$ -Notation siehe Infobox 13).

Die Quanten-Fouriertransformation (QFT, Quantum Fourier Transform) – unabhängig voneinander von Shor [A 5] und Coppersmith [A 44] entwickelt – ist das Analogon zur FFT auf dem Quantencomputer.

### Infobox 12: O-Notation

Eine Vergrößerung der Input-Datenmenge macht Probleme im Allgemeinen schwerer. Wie stark der Ressourcenbedarf eines Algorithmus dadurch wächst, charakterisiert dessen Effizienz. Üblicherweise wird diese Charakterisierung in der sogenannten Landau-Schreibweise („O-Notation“) angegeben. Für eine gegebene Problemgröße  $N$  gilt dabei:

Notation	Bedeutung	Anschauung
$\mathcal{O}(\log N)$	logarithmisches Wachstum	Aufwand wächst um konstanten Betrag, wenn $N$ sich verdoppelt
$\mathcal{O}(\log^m N)$	polylogarithmisches Wachstum	Aufwand wächst um (größeren) konstanten Betrag, wenn $N$ sich verdoppelt
$\mathcal{O}(\sqrt{N})$	Wachstum wie Wurzelfunktion	Aufwand verdoppelt sich, wenn $N$ sich vervierfacht
$\mathcal{O}(N)$	lineares Wachstum	Aufwand verdoppelt sich, wenn $N$ sich verdoppelt
$\mathcal{O}(N \log N)$	log-lineares/ quasi-lineares Wachstum	Für große $N$ gilt in guter Näherung: Aufwand verdoppelt sich, wenn $N$ sich verdoppelt
$\mathcal{O}(N^2)$	quadratisches Wachstum	Aufwand vervierfacht sich, wenn $N$ sich verdoppelt
$\mathcal{O}(N^m)$	polynomielles Wachstum	Aufwand wächst um $2^m$ -fache, wenn $N$ sich verdoppelt
$\mathcal{O}(2^N)$	exponentielles Wachstum	Aufwand verdoppelt sich, wenn $N$ sich um 1 erhöht

Genauer gesagt setzt die QFT die DFT auf den Amplituden eines Quantenzustandes um (zum Begriff »Amplitude« siehe Kapitel 1.2.1). Sie schreibt das Resultat dabei nicht in ein Ausgaberegister, sondern manipuliert den Eingabezustand direkt. Dazu benötigt sie lediglich  $\mathcal{O}(\log^2 N)$ <sup>6]</sup> Operationen, was eine exponentielle Beschleunigung im Vergleich zur FFT bedeutet.<sup>7</sup>

<sup>6</sup> Aus Kapitel 1.2 wissen wir, dass mit  $n$  Qubits  $2^n$  Amplituden verfügbar sind und somit  $N = 2^n$  Daten codiert werden können. Andersherum brauchen wir für  $N$  Daten also  $n = \log_2 N$  Qubits. Man kann äquivalent zu obigen Angaben die Anzahl der Rechenoperationen auch durch  $n$  ausdrücken. Dann haben wir für die FFT  $\mathcal{O}(n 2^n)$  und für die QFT  $\mathcal{O}(2^n)$ .

Bei der DFT erhalten wir aus den  $N$  Eingabedaten als Resultat  $N$  Ausgabedaten mit dem Ergebnis der Transformation. Im Gegensatz dazu liegt bei der QFT das Resultat, d. h. die transformierten Daten, als Amplituden eines Quantenzustandes vor und ist damit nicht direkt zugänglich (siehe Kapitel 1.2.3.3). Die QFT ist entsprechend nicht unbedingt als schnellerer Ersatz der FFT zu sehen, vielmehr bekommt sie ihre Bedeutung durch ihren weitverbreiteten Einsatz als Subroutine von anderen Algorithmen [A 4]. In [A 14] wird die QFT gar als die wichtigste Quantensubroutine überhaupt bezeichnet.

Quellen: [A 3, Kapitel 5.1], [A 4, Kapitel 7.8]

### 1.4.3 Algorithmen für fehlerkorrigierte Quantencomputer

#### 1.4.3.1 Shor-Algorithmus

Der 1994 vorgeschlagene Shor-Algorithmus [A 5], [A 6] war der erste Quantenalgorithmus, welcher eine Beschleunigung für ein anwendungsrelevantes Problem bot. Er kommt zur Anwendung bei der Faktorisierung großer Zahlen (und Berechnung von diskreten Logarithmen), wie sie in weitverbreiteten kryptographischen Verfahren wie RSA oder elliptische Kurven benutzt wird. Die Eingabe ist eine  $n$  Bit Zahl (damit können ganze, nichtnegative Zahlen bis maximal  $N = 2^n - 1$  dargestellt werden), die faktorisiert werden soll. Als Ergebnis erhalten wir in  $\mathcal{O}(n^3)$  Operationen einen nicht-trivialen Faktor der Zahl. Verbesserungen des Shor-Algorithmus skalieren sogar mit  $\mathcal{O}(n^2 \log n)$ . Der beste bekannte klassische Algorithmus, der Zahlkörpersieb (englisch: number field sieve), skaliert hingegen mit  $\mathcal{O}(e^{n^{1/3}})$  [A 4], [A 7]. Dies bedeutet, dass der Shor-Algorithmus eine exponentielle Beschleunigung liefert. Ein wichtiger Bestandteil und Grund für diese exponentielle Beschleunigung ist die QFT, welche im Shor-Algorithmus verwendet wird. Die Bedeutung und Konsequenzen sowie die Hardwareanforderungen des Shor-Algorithmus für die Kryptographie werden in [A 8] detailliert erläutert.

Weitere Quellen: [A 4], [A 7], [A 8], [A 9]

#### 1.4.3.2 Grover-Algorithmus

Neben dem Shor-Algorithmus ist der 1997 in [A 10] präsentierte Grover-Algorithmus einer der bekanntesten Quantenalgorithmus. Ursprünglich wurde er formuliert zur Durchsuchung von unstrukturierten Datenbanken. Nehmen wir als einfaches Beispiel zur Illustration dieser Situation an die Datenbank enthält  $N$  Einträge, wovon genau einer der gesuchte ist. Weiter nehmen wir an, dass es eine Subroutine gibt, welche zu jedem Eintrag entscheiden kann, ob es sich um den gesuchten Eintrag handelt oder nicht (beispielsweise indem gewisse Bedingungen überprüft werden). Die Implementierung der Subroutine spielt dabei für die Funktionalität des Algorithmus keine Rolle und wird

deshalb als Orakel bezeichnet (siehe Abbildung 20). Der bestmögliche klassische Algorithmus zum Finden des gesuchten Eintrags benötigt  $\mathcal{O}(N)$  Aufrufe des Orakels<sup>8</sup>. Der Grover-Algorithmus liefert jedoch bereits nach  $\mathcal{O}(\sqrt{N})$  Aufrufen des Orakels mit hoher Wahrscheinlichkeit den gesuchten Eintrag. Dies entspricht einer quadratischen Beschleunigung gegenüber dem klassischen Algorithmus.

Der Grover-Algorithmus funktioniert ebenfalls in der Situation, dass es mehrere gesuchte Einträge gibt. Auch gibt es Variationen des Grover-Algorithmus, welche nicht nur mit hoher Wahrscheinlichkeit, sondern mit Sicherheit einen gesuchten Eintrag liefern. Im Gegensatz zum Shor-Algorithmus ist der Grover-Algorithmus leichter zugänglich: er kann anschaulich geometrisch interpretiert und visualisiert werden. Auch wenn die quadratische Beschleunigung im Gegensatz zum klassischen Algorithmus nicht dramatisch wirkt, kann sie in der Anwendung große Auswirkungen haben. Beispielsweise ist die FFT auch lediglich quadratisch schneller als eine naive Implementierung der DFT, für die Praxis jedoch wird diese Beschleunigung als sehr signifikant betrachtet [A 4].

Beim Grover-Algorithmus ist der Fall jedoch komplizierter, da für eine reale Beschleunigung noch der Zusatzaufwand für Fehlerkorrektur berücksichtigt werden muss (siehe 1.5).

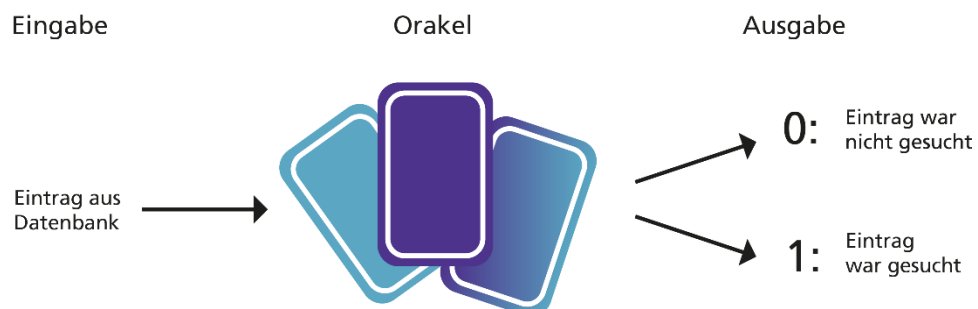


Abbildung 20: Der Grover-Algorithmus benötigt ein Orakel mit folgender Funktionalität: Als Eingabe erhält es einen Eintrag aus der Datenbank und als Ausgabe liefert es ein binäres Resultat: 1, falls es sich um einen gesuchten Eintrag aus der Datenbank gehandelt hat und 0, falls nicht.

Zu erwähnen ist, dass der Vorteil des Grover-Algorithmus gegenüber klassischen Algorithmen davon abhängt, dass das Orakel günstig zu berechnen ist. Sollte dies nicht der Fall sein, könnte dies Laufzeitvorteile konterkarieren. Auch sind Datenbanken in der Anwendung zumeist nicht unstrukturiert, sondern im Gegenteil sehr strukturiert. Diese können auch mit klassischen Algorithmen schneller durchsucht werden. Für geordnete

<sup>8</sup> Da es sich um eine unsortierte Datenbank handelt, bleibt auf dem klassischen Computer nichts anderes übrig als ein Eintrag nach dem anderen zu überprüfen. Dadurch würde man im Mittel nach  $N/2$  Überprüfungen (d. h. Aufrufen des Orakels) den gesuchten Eintrag finden. Die Anzahl der Orakelaufufe wachsen also linear mit  $\mathcal{O}(N)$ .



Daten ist höchstens ein konstanter Faktor an Beschleunigung durch einen Quantenalgorithmus gegenüber einem optimalen klassischen Algorithmus möglich. Somit ist durch den Grover-Algorithmus keine Beschleunigung von Datenbankabfragen oder Internetsuchanfragen zu erwarten [A 4], [A 11].

In [A 12] wurde bewiesen, dass der Grover-Algorithmus optimale Komplexität hat. Für die gegebene Aufgabe kann es keinen Algorithmus geben, der besser als  $\mathcal{O}(\sqrt{N})$  skaliert. Wie die QFT findet der Grover-Algorithmus Anwendung als Subroutine in komplexeren Quantenalgorithmen, z. B. in Optimierungsanwendungen oder der Graphentheorie, siehe z. B. [A 13], [A 14, Tabelle 2].

Weitere Quellen: [A 4, Kapitel 9], [A 9], [A 15]

Algorithmen zur Lösung linearer Gleichungssysteme spielen eine herausragende Rolle bei der Modellierung und Simulation von Problemen, Effekten und Phänomenen verschiedenster Fachgebiete. Sie tauchen in natürlicher Weise unter anderem in der Physik und der Chemie, den Ingenieurs- und Computerwissenschaften, der Finanz- und Versicherungsbranche sowie dem maschinellen Lernen auf. Das Ziel ist dabei zu einer gegebenen  $N \times N$  Matrix  $A$  und einem gegebenen Vektor  $b$  einen Vektor  $x$  zu finden, sodass  $Ax = b$  gilt. Der beste bekannte klassische Algorithmus, der dieses Problem (näherungsweise und für eine bestimmte Klasse an Matrizen) löst, ist der CG-Algorithmus (conjugate gradient, deutsch: Verfahren der konjugierten Gradienten). Dieser hat eine Laufzeitkomplexität von  $\mathcal{O}(Nsk \log(\frac{1}{\epsilon}))$ , wobei  $s$  die Sparsity (Maß für die Dünnbesetztheit) und  $\kappa$  die Kondition (Empfindlichkeit gegen Störungen) der Matrix  $A$  sind und  $\epsilon$  die gewünschte Genauigkeit der Näherungslösung beschreibt.

Der von Harrow, Hassidim und Lloyd vorgeschlagene (und nach den Autoren benannte) HHL-Algorithmus [A 16] löst das obige lineare Gleichungssystem  $Ax = b$  in dem Sinn, dass er ein Quantenzustand  $|\tilde{x}\rangle$  berechnet, der bis auf Genauigkeit  $\epsilon$  die »Quantenversion«  $|x\rangle$  von  $x$  approximiert. Die Laufzeitkomplexität beträgt dabei  $\mathcal{O}(\log(N) s^2 \kappa^2 \frac{1}{\epsilon})$ , was einer exponentiellen Beschleunigung in der Anzahl der Inputdaten  $N$  gegenüber dem CG-Algorithmus entspricht. Allerdings ist zu beachten, dass die Skalierung in den anderen Parametern schlechter ist, was einen Geschwindigkeitsvorteil des Quantenalgorithmus gegenüber dem klassischen Algorithmus wieder zunichte oder Einschränkungen auf spezielle Problemklassen erforderlich machen kann. Weiterhin ist zu beachten, dass der HHL-Algorithmus Zugriff auf den Eingabevektor  $b$  in Form eines Quanten-Zustandes  $|b\rangle$  benötigt. Dieser muss effizient auf dem Quantencomputer zu erzeugen sein, um einen möglichen Vorteil des HHL-Algorithmus nicht zu unterminieren. Ebenso benötigt der HHL-Algorithmus Zugriff auf  $A$ , der ebenfalls effizient umsetzbar sein muss. Auch erhalten wir, wie oben erwähnt, nicht den vollen Lösungsvektor  $x$ , sondern ein Quantenzustand  $|\tilde{x}\rangle$ , dessen Amplituden die Komponenten von  $x$  approximieren. Durch  $\mathcal{O}(N)$  Messungen (und

entsprechend vielen Durchläufen des HHL-Algorithmus) könnte zwar der Vektor  $\tilde{x}$  gewonnen werden, jedoch würde dies jeden Laufzeitvorteil zerstören. Dieses Problem tritt nicht auf, wenn das zugrunde liegende Anwendungsproblem nur spezifische Stichproben von  $|\tilde{x}\rangle$  benötigt.

Es existieren diverse Verbesserungs- und Generalisierungsvorschläge des HHL-Algorithmus, die die Laufzeitkomplexität (zum Teil unter erweiterten Annahmen) verbessern oder zu deterministischen Ergebnissen führen [A 17], [A 18], [A 19], [A 20]. Diese werden oftmals unter der Bezeichnung QLSA (Quantum Linear System Algorithm) geführt. Viele weitere Quantenalgorithmen, beispielsweise aus dem maschinellen Lernen und der Lösung von Differentialgleichungen, nutzen bzw. basieren auf dem HHL-Algorithmus und seinen Fortentwicklungen [A 45].

Weitere Quellen: [A 7], [A 21], [A 22], [A 46]

#### 1.4.4 Variationelle Quantenalgorithmen

Variationelle Quantenalgorithmen (VQA, englisch: variational quantum algorithms) haben sich als die führende Strategie etabliert, um in der NISQ Ära Quantenanwendungen umzusetzen. Sie zeichnen sich durch ihren hybriden Ansatz aus, bei dem klassischer und Quantencomputer zusammenarbeiten. Hierbei werden parametrisierte Schaltkreise auf dem Quantencomputer ausgeführt, die ihre Parameter durch ein Optimierungsverfahren auf einem klassischen Computer erhalten. Dies erlaubt die Konstruktion von Algorithmen, die lediglich Quantenschaltkreise mit geringer Tiefe benötigen. Dadurch sind sie für den Einsatz auf der aktuellen NISQ-Hardware mit den zugehörigen Beschränkungen besonders geeignet. Entsprechend wurden sie für (fast) jede Anwendung, die für Quantencomputing in Frage kommt, entwickelt (siehe Abbildung 21).

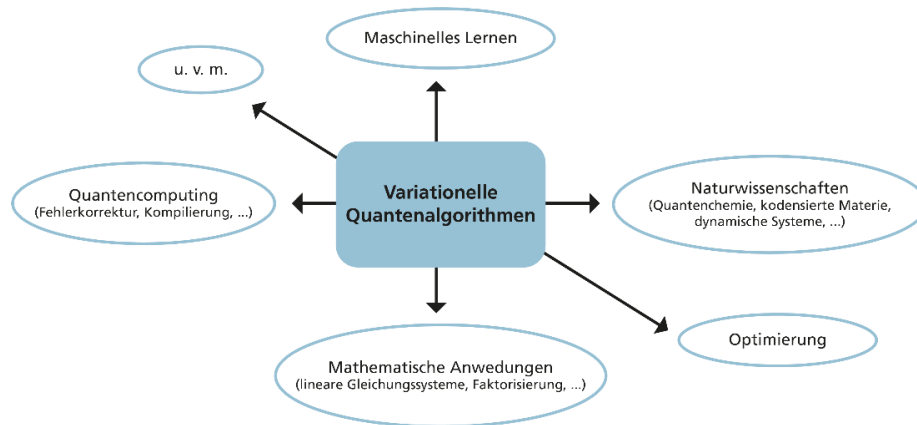


Abbildung 21: Anwendungsgebiete variationeller Quantenalgorithmen, siehe [A 28]

Bei variationellen Quantenalgorithmen wird das zu lösende Problem über eine Zielfunktion dargestellt. Diese könnte beispielsweise die Kosten oder den  $\text{CO}_2$ -Ausstoß bei verschiedenen Energieanteilen an einem Strommix oder den Profit bei verschiedenen Gewichtungen von Aktien in einem Portfolio modellieren. Mögliche Eingabedaten für die Zielfunktion kann man parametrisieren und somit systematisch in die Zielfunktion einsetzen und diese auswerten. Finden wir den Satz an Parametern, an dem die Zielfunktion ihr Minimum bzw. Maximum annimmt, haben wir die Lösung des zugrundeliegenden Problems gefunden. Dies kann realisiert werden, indem ausgehend von einem Satz von Anfangsparametern diese iterativ so variiert werden, dass der Wert der Zielfunktion nach jeder Iteration verbessert wird. Das setzt voraus, dass die Zielfunktion in jedem Iterationsschritt ausgewertet wird, um die Variation der Parameter für die nächste Iteration zu bestimmen. Auf klassischen Computern ist dies oft nicht effizient möglich.

Hier setzen VQA an. Bei ihnen wird ein als Ansatz (oder variationelle Form) bezeichneter parametrisierter Quantenschaltkreis gewählt, um verschiedene Eingabedaten für die Zielfunktion zu erzeugen. Ausführung und Messung des Ansatzes liefert die Auswertung der Zielfunktion, deren Berechnung auf den Quantencomputer verlagert wurde. Das Ergebnis wird anschließend auf dem klassischen Computer verwendet, um die optimierten Parameter für die nächste Iteration zu berechnen. In Abbildung 22 ist der Ablauf nochmals zusammengefasst. Die Iterationsschleife wird abgebrochen, sobald die Qualität des Ergebnisses gut genug ist oder andere Abbruchbedingungen erfüllt sind. Die Wahl eines guten Ansatzes ist entscheidend für VQA und kann nach verschiedenen Kriterien getroffen werden, z. B. problemspezifisch oder hardwarespezifisch (siehe [A 24] für weitere Beispiele). Da dies ein komplexes Problem ist, sind die meisten Ansätze heuristischer Natur, sodass allgemeine Aussagen über Laufzeitverbesserungen nicht möglich sind.

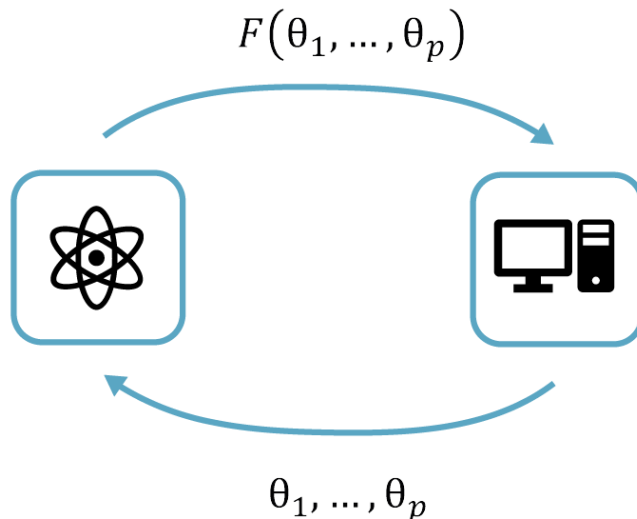


Abbildung 22: Funktionsweise eines variationellen Quantenalgorithmus: Für ein Satz an Parametern  $\theta_1, \dots, \theta_p$  liefert der Quantencomputer die Auswertung  $F(\theta_1, \dots, \theta_p)$  der Zielfunktion. Auf dem klassischen Computer wird daraus der nächste zu überprüfende Parametersatz  $\theta_1, \dots, \theta_p$  berechnet und wieder an den Quantencomputer übergeben. Dies wird so lange iteriert bis ein optimaler Satz an Parametern erzeugt wurde oder andere Abbruchkriterien eintreten.

Weitere Quellen: [A 7, Kapitel 3.3.1], [A 23], [A 24], [A 25], [A 26], [A 27]

#### 1.4.4.1 QAOA

Der bekannteste VQA für klassische Optimierungsprobleme ist der Quantum Approximate Optimization Algorithm (QAOA), der in [A 29] zur approximativen Lösung von kombinatorischen Optimierungsproblemen vorgeschlagen wurde. Diese äußerst wichtige Klasse von Problemen taucht in beinahe jedem Anwendungsgebiet auf, unter anderem in der Logistik, der Produktionsplanung, im Netzwerkdesign, der Verkehrsplanung wie auch in der OperationsResearch (siehe Use Cases 2.2). Aufgrund der Größe von realistischen Problemen ist es oftmals nicht möglich eine optimale Lösung zu finden. Somit sind approximative Algorithmen (wie QAOA), die Näherungen an die optimale Lösung berechnen, von hoher praktischer Relevanz.

Um für QAOA zugänglich zu sein, muss ein gegebenes kombinatorisches Optimierungsproblem erst als QUBO (quadratic unconstrained binary optimization) Problem umformuliert werden. Für eine Vielzahl an Beispielen liegt diese Umformulierung bereits vor [A 30], [A 31], [A 32], [A 33], [A 34]. Aus der QUBO-Form lässt sich dann die Zielfunktion herleiten. Der QAOA-Ansatz besteht nun aus zwei alternierenden parametrisierten Quantenoperationen: die erste enthält die Zielfunktion, während die zweite Überlagerungen herstellt (diese Operation wird als Mixer bezeichnet). Wie bei allen VQA wird das Resultat verwendet, um mit einem klassischen Optimierer verbesserte Parameter zu berechnen. Dies erweist sich als schwieriges Problem, da die

Optimierungslandschaft von QAOA nicht-konvex mit vielen lokalen Minima ist [A 35] Entsprechend wurde und wird großer Aufwand in das Finden guter klassischer Optimierer gesteckt. Es existieren Verallgemeinerungen von QAOA zum Quantum Alternating Operator Ansatz (wiederum abgekürzt mit QAOA) [A 36], der die Alternierung von allgemeineren Quantenoperationen erlaubt und Vorteile z. B. bei Optimierungsproblemen mit Nebenbedingungen bietet. Außerdem gibt es den sogenannten warm-starting QAOA [A 37] der QAOA von einer klassisch berechneten Approximation der Lösung startet.

Weitere Quellen: [A 9], [A 23], [A 24], [A 38]

#### 1.4.4.2 VQE

Für eine quadratische Matrix  $A$  nennt man die Zahl  $\lambda$  Eigenwert und den Vektor  $x$  Eigenvektor, falls gilt:  $Ax = \lambda x$ . Das Finden der Eigenwerte und Eigenvektoren wird als Eigenwertproblem bezeichnet. Die Lösung von hochdimensionalen Eigenwertproblemen ist dabei zentral für verschiedenste Anwendungen, wie z. B. dem Ranking von Suchmaschinen [A 39], der Strukturdynamik, in elektrischen Netzwerken oder der Makroökonomie [A 40]. Insbesondere in der Simulation quantenchemischer und quantenphysikalischer Phänomene spielen Eigenwertprobleme eine große Rolle. Allerdings sind diese bereits für kleine Systeme in klassischen Computer schwer bis gar nicht mehr lösbar. Bessere Algorithmen würden hier nicht nur einen großen Fortschritt akademischer Natur bedeuten, sondern wären direkt auch in Anwendungsbereichen wie beispielsweise der Medizin (z. B. für die Entwicklung neuer Medikamente) und den Materialwissenschaften (z.B. für neue Batterien) einsetzbar [A 28], [A 41].

Der Variational Quantum Eigensolver (VQE) [A 42], [A 43] wurde entwickelt, um die Grundzustandsenergie von Quantensystemen wie z. B. von Festkörpersystemen oder Molekülen zu berechnen (dieses Auffinden wird auch als »electronic structure problem« bezeichnet). Daraus lassen sich u. a. Reaktionen, stabile Strukturen und optische Eigenschaften voraussagen [A 41]. Mathematisch entspricht dies dem Lösen eines Eigenwertproblems, genauer gesagt dem Finden des kleinsten Eigenwerts einer hermiteschen Matrix  $H$ . Zielfunktion ist dabei der Energieerwartungswert  $\langle H \rangle$ , dessen Minimierung den kleinsten Eigenwert liefert (Rayleigh-Ritz-Prinzip). Als variationeller Algorithmus bewerkstelligt VQE diese Minimierung in hybrider Form durch einen variationellen Ansatz, der die Auswertung der Zielfunktion auf dem Quantencomputer umsetzt und einem klassischen Optimierer, der die Parameter des Ansatzes bestimmt. Für den Ansatz gibt es eine Vielzahl an Vorschlägen, u. a. problemspezifische (z. B. aus der Quantenchemie), heuristische, problemagnostische Ansätze (für eine große Bandbreite an Problemen geeignet) oder hardware-spezifische Ansätze (gut implementierbare Quantenschaltkreise) [A 41]. Das macht VQE sehr flexibel einsetzbar, jedoch hängen sowohl Performance als auch Hardwareanforderungen stark von der Wahl

des Ansatzes ab. Weiterhin existieren eine Vielzahl an Adaptionen und Erweiterungen von VQE, die beispielweise angeregte Energiezustände berechnen können [A 23], [A 24].

Weitere Quellen: [A 23], [A 24], [A 26], [A 28], [A 41], [A 42]

Autor: Dr. Andreas Sturm (Fraunhofer IAO)

Kontakt: [andreas.sturm@iao.fraunhofer.de](mailto:andreas.sturm@iao.fraunhofer.de)

## 1.5 Exkurs: Quantenfehlerkorrektur

Wie in Kapitel 1.2.4.1 beschrieben, beruht der mögliche Vorteil eines Quantencomputers gegenüber klassischen Computern unter anderem darauf, dass sie mit Superpositionen aller  $2^n$  verschiedenen Basiszustände der  $n$  Qubits arbeiten können. Diese Superpositionen sind jedoch äußerst empfindlich gegenüber Störungen seitens der Umgebung. Da es in der Praxis unmöglich ist, ein Quantensystem vollständig von seiner Umgebung zu isolieren, muss man sich mit der Tatsache auseinandersetzen, dass die Quantenoperationen, welche auf den physikalisch realisierten Qubits durchgeführt werden, Fehlerraten aufweisen. Beispielsweise betragen diese Fehlerraten bei dem von der Fraunhofer-Gesellschaft genutzten IBM Quantum System One Ehningen momentan (Dezember 2022) ca. 0,02 % für 1-Qubit-Quantengatter und ca. 0,8 % für 2-Qubit-Quantengatter. Neben der beschränkten Anzahl von Qubits sind diese Fehler derzeit noch ein wesentliches Hindernis auf dem Weg zu anwendungsrelevantem Quantencomputing.

In klassischen Computern hingegen ist die Fehlerrate der elementaren Rechenoperationen heutzutage so gering ( $< 10^{-17}$ ; [E 1]) dass keine Fehlerkorrektur benötigt wird. Anders sieht es aus, wenn Daten über größere Entfernungen übertragen werden sollen. Hierbei können Fehler auftreten - z. B. ein Umspringen eines Bits von 1 auf 0. Das einfachste Korrekturverfahren besteht darin, jedes einzelne Bit nicht nur einmal, sondern mehrmals zu übertragen (also z.B. 000 statt 0 und 111 statt 1). Tritt ein einzelner Fehler auf (z.B. 111  $\rightarrow$  101), kann anhand der übrigen Bits der ursprüngliche Wert rekonstruiert werden. Die Fehlerwahrscheinlichkeit ändert sich dabei von  $p$  (für das Umspringen eines einzelnen Bits) auf  $3p^2 - 2p^3$  (für die Wahrscheinlichkeit, dass zwei von drei oder alle drei Bits umspringen). Sofern die Fehlerwahrscheinlichkeit eine gewisse Schwelle unterschreitet (hier:  $p = \frac{1}{2}$ ), wird auf diese Weise der Fehler insgesamt reduziert. Reicht dies noch nicht aus, kann der Fehler weiter unterdrückt werden, z. B., indem man jedes Bit mehr als drei Mal überträgt.

Auf ähnliche Weise können auch Quantencomputer korrigiert werden: mehrere fehlerbehaftete physikalische Qubits stellen hierbei ein weniger fehleranfälliges logisches Qubit dar. Durch die Messung sogenannter Fehlersyndrome wird festgestellt, ob ein

Fehler aufgetreten ist, und – falls ja – wie dieser wieder korrigiert werden kann. Im einfachsten Fall sind – ähnlich wie im oben erwähnten klassischen Fall – drei physikalische Qubits nötig, um eine Korrektur unerwünschter Bitflips ( $|0\rangle \leftrightarrow |1\rangle$ ) zu ermöglichen (Infobox 14). Neben Bitflips können Qubits im Gegensatz zu klassischen Bits jedoch auch von Phasenflips ( $|0\rangle + |1\rangle \leftrightarrow |0\rangle - |1\rangle$ ) betroffen sein. Eine Erweiterung des Codes auf mindestens fünf physikalische Bits [E 1] erlaubt es, auch diese Fehler zu korrigieren. Alle weiteren möglichen Fehler, die an einem einzelnen Qubit auftreten können, werden durch den mit der Messung der Fehlersynndrome einhergehenden Kollaps des Quantenzustands (siehe Abschnitt 1.2.3.3) auf Bit- bzw. Phasenflips reduziert und damit ebenfalls korrigiert.

Neben dem im Infobox 14 erläuterten Grundprinzip der Quantenfehlerkorrektur sind in der Realität noch weitere Aspekte zu beachten: So benötigen Kodierung, Messung und Korrektur zusätzliche Quantenoperationen, die ihrerseits auch fehlerbehaftet sind. Außerdem können sich ursprünglich lokal (d.h. an einzelnen Qubits) auftretende Fehler im Laufe der Rechnung auf weitere Qubits ausdehnen. Insgesamt dürfen also bei der Implementierung der Quantenfehlerkorrektur nicht mehr zusätzliche Fehler eingeführt als am Ende korrigiert werden. Nach der Theorie des **fehlertoleranten Quantencomputings** [E 2], [E 3] ist dies möglich, sofern die physikalischen Fehlerraten eine bestimmte Schwelle unterschreiten und dabei gewisse Annahmen erfüllt sind, beispielsweise dass Fehler nur an einzelnen Qubits oder Gruppen benachbarter Qubits (mit hinreichend schnell abfallenden Korrelationen) auftreten. Die Größe der Fehlerschwelle hängt vom verwendeten Quantenfehlerkorrekturcode ab: nach gegenwärtigem Stand der Forschung verhalten sich hier die **Surface Codes** am günstigsten mit einer geschätzten Fehlerschwelle von ca. 1 % [E 4]. Je nachdem, wie weit der tatsächliche Fehler unterhalb dieser Schwelle liegt, wird eine unterschiedlich große Zahl physikalischer Qubits benötigt, um ein fehlerfreies logisches Qubit zu kodieren. Bei einer physikalischen Fehlerrate von höchstens 0,1 % wird dieser Aufwand auf ca. 1.000 bis 10.000 physikalische Qubits pro logischem Qubit geschätzt [E 5]. Zu beachten ist, dass neben den Quantengattern auch der Messvorgang zur Bestimmung der Fehlersynndrome schnell und präzise funktionieren muss, damit Fehler während der Rechnung rechtzeitig erkannt und korrigiert werden können. Auch diese Bedingungen werden von der gegenwärtigen IBM-Quantenhardware noch nicht erfüllt: der Auslesefehler beträgt ca. 1 % statt der benötigten 0,1 % bei einer Dauer von 0,8  $\mu$ s im Vergleich zu 0,3  $\mu$ s für ein CNOT-Gatter (1  $\mu$ s = 1 Mikrosekunde = 0,000001 Sekunden).

### Infobox 13: Drei Qubit Bitflip Code

Der einfachste Quantenfehlerkorrekturcode ist in der Lage Bitflips auf einzelnen Qubits zu korrigieren. Der Zustand  $|\psi\rangle = a|0_L\rangle + b|1_L\rangle$  eines logischen Qubits wird hierbei wie folgt in drei physikalischen Qubits kodiert:  $|\psi\rangle = a|000\rangle + b|111\rangle$ , also:

$$\begin{aligned} |0\rangle &\rightarrow |0_L\rangle = |000\rangle \\ |1\rangle &\rightarrow |1_L\rangle = |111\rangle \end{aligned}$$

Nehmen wir nun an, dass bei jedem Qubit mit Wahrscheinlichkeit  $p$  ein Bitflip auftritt. Dieser Fehler kann dann wie folgt korrigiert werden: Zunächst wird durch geeignete Messungen bestimmt, welches Qubit von einem Bitflip betroffen ist. Anschließend wird das entsprechende Qubit zurückgeflippt und damit der ursprüngliche Zustand  $|\psi\rangle$  wiederhergestellt. Eine wichtige Rolle spielen hierbei die besonderen Eigenschaften des quantenmechanischen Messprozesses. Anders als im klassischen Fall darf man beispielsweise nicht einfach den Zustand aller drei Qubits direkt vermessen, da aufgrund des damit einhergehenden Zustandskollaps der ursprüngliche Superpositionszustand zerstört wird (siehe Abschnitt 1.2.3.3). Stattdessen führt man sogenannte kollektive Messungen an zwei Qubits gleichzeitig durch. Die Messung des Operators  $Z_1Z_2$  beispielsweise liefert das Ergebnis „+1“ oder „-1“ je nachdem, ob sich die Qubits 1 und 2 im selben Zustand oder in unterschiedlichen Zuständen befinden, ohne jedoch preiszugeben, um welche Zustände es sich hierbei handelt. Diese kollektive Messung wird an einem zusätzlichen Hilfsqubit durchgeführt, welches zuvor durch jeweils ein CNOT-Gatter an die Qubits 1 bzw. 2 gekoppelt worden ist. Nach einer weiteren Messung ( $Z_2Z_3$ ) an Qubits 2 und 3 kann man gemäß folgender Tabelle bestimmen, welches Qubit von einem Bitflip betroffen ist, und den Zustand entsprechend korrigieren:

Fehlersyndrom		Korrektur
$Z_1Z_2$	$Z_2Z_3$	
+1	+1	keine
+1	-1	$X_3$ (flip Qubit 3)
-1	+1	$X_1$ (flip Qubit 1)
-1	-1	$X_2$ (flip Qubit 2)

Wie im klassischen Fall (siehe Haupttext) funktioniert die Korrektur nur dann, wenn höchstens ein Qubit von einem Bitflip betroffen ist. Insgesamt wird hierdurch die Fehlerwahrscheinlichkeit von  $p$  auf  $3p^2 - 2p^3$  reduziert.

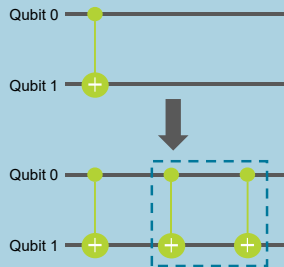


Eine für die Zukunft des Quantencomputings entscheidende Frage ist, ob Quantenfehlerkorrektur auch bei physikalisch realistischen Fehlern funktionieren wird, welche die oben erwähnten Modellannahmen der Schwellentheoreme (beispielsweise bzgl. abfallender Korrelationen) möglicherweise nicht vollständig erfüllen. Bevor diese Frage geklärt werden kann, wird noch einige Zeit vergehen, in der die Quantenhardware weiter verbessert werden muss, um die Fehlerraten deutlich unter die Fehlerschwelle zu drücken und die Zahl der physikalischen Qubits bis auf ca. 100.000 bis 1.000.000 zu vergrößern. Innerhalb der nächsten Jahre werden wir es daher mit noch nicht (vollständig) fehlerkorrigierten sogenannten **NISQ** (Noisy Intermediate-Scale Quantum) Computern zu tun haben [E 6], bei denen die Länge der durchführbaren Quantenschaltkreise aufgrund der physikalischen Fehlerraten beschränkt ist. Für diese gibt es eine Reihe von schon heute einsetzbaren **Fehlermitigationsverfahren** (siehe **Infobox 15**) die – im Gegensatz zur oben diskutierten Quantenfehlerkorrektur – ohne zusätzliche Qubits auskommen und Fehler bis zu einem gewissen Maß reduzieren, jedoch nicht vollständig unterdrücken können können.

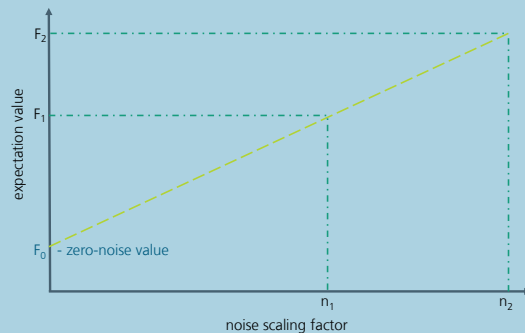
Zu nennen sind hier die Mitigation von zuvor charakterisierten Auslesefehlern durch klassische Nachverarbeitung der experimentellen Auslesestatistik [E 7] und die Zero Noise Extrapolation, bei der die Fehler zunächst absichtlich verstärkt werden, um danach eine Extrapolation auf Fehlerstärke Null durchzuführen [E 8]. Eine spannende und gegenwärtig noch offene Frage ist, ob es bestimmte Quantenalgorithmen gibt, die ein gewisses Maß von Fehlern verkraften und auf diese Weise auch schon in näherer Zukunft einen anwendungsrelevanten Vorteil gegenüber klassischen Computern erzielen können.

Nachdem in diesem Kapitel die Grundlagen des Quantencomputings gelegt wurden, werden im nächsten Kapitel konkrete Anwendungsfälle besprochen.

### Infobox 14: Fehlermitigationsverfahren

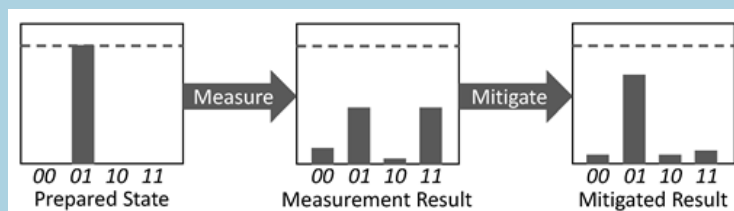


a) Fehlerverstärkung durch zusätzliche Gatter



b) Extrapolation zum fehlerfreien Wert

Da die Anwendbarkeit von Quantenalgorithmen auf den derzeit zu Verfügung stehenden NISQ-Computern (insbesondere auch dem IBM-Quantencomputern) wesentlich durch die auf der Quantenhardware auftretenden Fehler beschränkt ist, kommt der Entwicklung und Analyse verschiedener Strategien zur Mitigation (Abschwächung) dieser Fehler und somit zur Steigerung der Zuverlässigkeit von Quantensoftware-Komponenten eine wesentliche Bedeutung zu. Zur Mitigation von Fehlern beim Durchführen von Quantengattern wird deshalb einerseits das Verfahren der Richardson-Fehlerextrapolation bzw. Zero Noise Extrapolation implementiert, welches auf der Fähigkeit beruht, die Stärke der Fehler durch zusätzliche Quantengatter zu variieren. Andererseits werden zur Korrektur von Auslesefehlern verschiedene Matrixinversions-, sogenannte Unfolding-Methoden, verwendet. Dabei wird eine Kalibrationsmatrix gemessen in der Informationen über die Messfehler enthalten sind. Diese wird mit der theoretischen, fehlerfreien Matrix verglichen um im Anschluss die Messwerte mitigieren zu können [E 9].



Autor: Dr. Thomas Wellens (Fraunhofer IAF)

Kontakt: [thomas.wellens@iaf.fraunhofer.de](mailto:thomas.wellens@iaf.fraunhofer.de)

## 2 DIE SEQUOIA ANWENDUNGSFÄLLE

### 2.1 Das assoziierte Unternehmensnetzwerk

Das Interesse der Wirtschaft an Quantentechnologien, insbesondere am Quantencomputing wächst in den letzten Jahren rapide und stetig (siehe Kapitel 1.1). Durch den industriellen Einsatz von Quantencomputern versprechen sich Unternehmen perspektivisch enormes wirtschaftliches Potenzial. In Vorbereitung auf die konkreten SEQUOIA Anwendungsfälle, welche zentraler Gegenstand dieses Kapitels sind, wird zunächst ein Überblick dieser Unternehmen respektive der relevanten Branchen und potenziellen Anwendungsgebiete gegeben.

Das assoziierte Unternehmensnetzwerk im SEQUOIA Projekt besteht aus über 30 Unternehmen aus unterschiedlichen Branchen. Darunter sind nicht nur produzierende und entwickelnde Gewerbe – auch für Dienstleistungsunternehmen oder Verwaltungen ist die Zukunftstechnologie von hoher Relevanz. Neben Firmen aus den Bereichen Automotive, Transport und Mobilität, dem Chemie- und Pharmaziesektor sowie Unternehmen aus der Energie- und Finanzwirtschaft spielt Quantencomputing insbesondere für innovationsfokussierte Technologieanwender eine bedeutende Rolle. Das Netzwerk bringt mittels regelmäßiger Netzwerktreffen ihre Anwendungsfälle und ihr branchenspezifisches Domänenwissen im Quantencomputing-Kontext ein. Damit wird eine fortwährende industrielle Relevanz für die zu erforschenden Quantenanwendungen sichergestellt.

Einzelne Firmen des Netzwerkes planen bereits kurz- und mittelfristige Projekte im Bereich Quantencomputing. Deren zentrale Motivation ist durch effizientere Prozessoptimierungen das Einsparen verschiedenster Ressourcen zu ermöglichen.

Die konkreten Anwendungsfälle des Unternehmensnetzwerkes im SEQUOIA Projekt definieren die jeweiligen Quantenumsetzungen der Forschungspartner. Dazu zählen unter anderem kombinatorische Optimierungsprobleme in der Elektromobilität und der verarbeitenden Industrie, die Lösung von (gekoppelten) Differentialgleichungen zur Unterstützung von CFD-Berechnungen (Computational Fluid Dynamics) oder die Lösung von Varianten des TSP-Problems (Travelling Sales Person) im Bereich der Logistik zur nachhaltigen Optimierung von Routing-Problemen.

Gegenstand der folgenden Unterkapitel (2.2.1 - 2.2.6) ist die Sammlung und detaillierte Untersuchung hybrider Quantenalgorithmen für diese Anwendungsfälle. Eine genaue Analyse und Bewertung hinsichtlich praktischer Relevanz geschieht anhand von repräsentativen Fokus-Anwendungen. Speziell für diese werden mögliche Lösungsansätze vorgestellt und hinsichtlich ihrer Anwendbarkeit beurteilt. Es ist

hervorzuheben, dass die vorgestellten Anwendungsfälle durch interaktive Jupyter Notebooks gestützt werden, um Entwickelnden niederschwellig erste Hands-on Erfahrungen zu ermöglichen und einen fundierten Wissenstransfer sicherzustellen.

Autorin: Chiara Stephan (Fraunhofer IAO)

Kontakt: [chiara.stephan@iao.fraunhofer.de](mailto:chiara.stephan@iao.fraunhofer.de)

## 2.2 Anwendungsfälle

### 2.2.1 Fokusanwendung 1 »Optimierung von Ladeplänen einer gegebenen Ladesäuleninfrastruktur«

Im Zuge des Verbundprojekts »**LamA (Laden am Arbeitsplatz)**« wird seit Juli 2018 bundesweit an 37 Fraunhofer-Instituten **Ladeinfrastruktur für Elektrofahrzeuge** errichtet [UC 1]. Ziel dieses Use Cases war es Ladepläne für diese Infrastruktur zu optimieren. Zwei beispielhafte Optimierungen wären zum Beispiel basierend auf Wetterprognosen die Elektrofahrzeuge dann zu laden, wenn viel erneuerbare Energie (Sonne, Wind, etc.) verfügbar ist. Oder basierend auf Strompreisprognosen zu den Tageszeiten zu laden, wann der Strom am günstigsten ist. Beide Beispiele basieren im Wesentlichen darauf, dass die E-Autos am Arbeitsplatz geladen werden und somit Ankunfts- und Abfahrtszeiten (ungefähr) bekannt sind und eine große Zeitspanne (ca. acht Stunden) zum Laden zur Verfügung steht. Viele weitere Optimierungen sind denkbar, in Anbetracht der vielen Einschränkungen von den derzeitigen NISQ-Computern wurde in diesem Use Case jedoch ausschließlich der Aspekt der Lastspitzenreduktion betrachtet. Diese reduziert Kosten, schont die Lade- und Netzinfrastruktur und ist somit ein erster Schritt in Richtung einer ganzheitlichen Ladeplanoptimierung. Als konkrete Aufgabe wurde definiert die Spitzenlast, welche aus dem Stromnetz gezogen wird, zu minimieren. Dabei ist gegeben wie viele Elektrofahrzeuge an einer Ladesäule sind, wann sie ankommen sowie abfahren und wie viel Energie sie laden müssen. Abbildung 23 veranschaulicht die Situation. Man kann erkennen, dass noch weitere Vereinfachungen wie z.B. eine Diskretisierung der Zeitachse, die Beschränkung auf diskrete Ladestufen und kleine Problemstellungen getroffen wurden. Dies hat den Hintergrund NISQ-kompatible Probleme zu erhalten, welche jedoch noch den Kern des zugrunde liegenden Use Cases modellieren. Es handelt sich somit um Proof-of-Concept-Modelle. Im Folgenden werden die Schritte zur Umsetzung als Quanten-Anwendung und die damit erzielten Resultate vorgestellt.

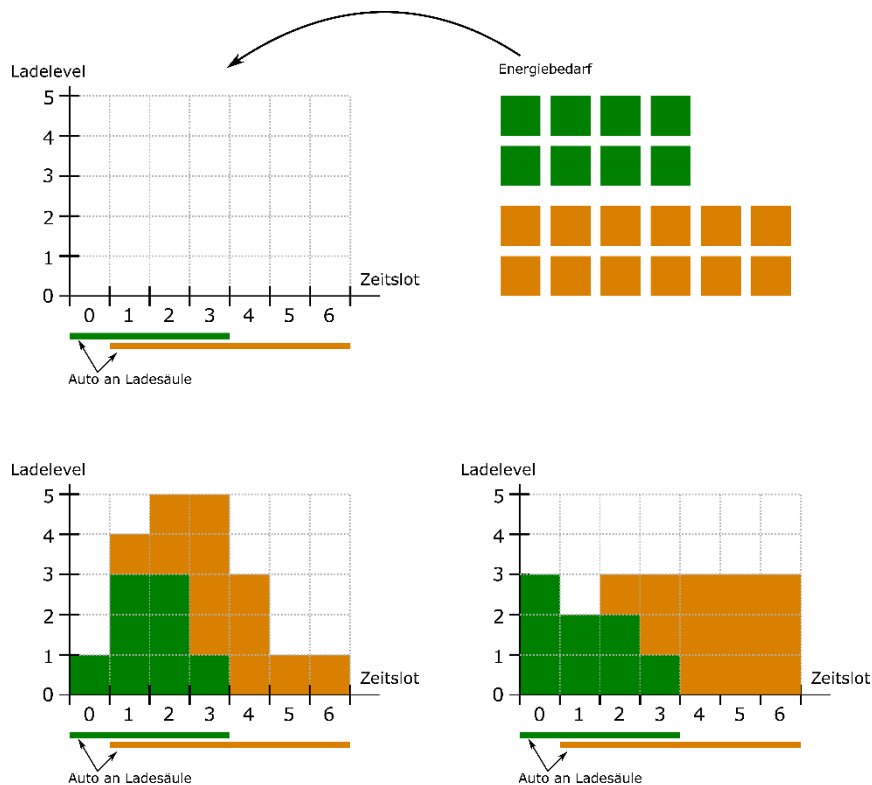


Abbildung 23: Proof-of-Concept-Modell zum Fokus-Anwendungsfall »Optimierung von Ladeplänen«. (Beispiel für abstrakte Beschriftung: 1 Zeitslot = 1 Stunde, 1 Ladelevel = 5 kW).

Als ersten Schritt zur Umsetzung als Quanten-Anwendung wurde eine mathematische Modellierung vorgenommen. Diese resultierte in einem quadratischen, ganzzahligen Optimierungsproblem mit Nebenbedingungen. Zugehörig wurde eine objekt-orientierte Implementierung in der Programmiersprache Python erarbeitet, welche Verständlichkeit, Wartbarkeit und insbesondere Erweiterbarkeit auf spätere komplexere Modelle sicherstellt. Basierend auf dem Modul `Qiskit Optimization` wurde eine Schnittstellenfunktion programmiert, welche Instanzen dieser Implementierung in die Welt des Quantensoftware-Framework `Qiskit` überträgt. Ein weiterer notwendiger Schritt, um für den Quantenalgorithmus QAOA (siehe Abschnitt QAOA) zugänglich zu sein, ist die Transformation des obigen mathematischen Modells in ein quadratisches, binäres Problem ohne Nebenbedingung (QUBO, englisch für quadratic unconstrained binary optimization problem). Alle diese klassischen Vorarbeiten sind in [UC 2] zu finden.

Als nächster Schritt wurde der Quantenalgorithmus QAOA zuerst als geeigneter Algorithmus für das hier betrachtete Problem identifiziert, anschließend im Detail analysiert sowie auf verschiedenen Abstraktionsstufen implementiert und auf Simulatoren ausgeführt. Ein wichtiges Resultat aus diesem Arbeitsschritt ist die Erkenntnis, dass eine Implementierung hoher Abstraktion, wie sie die `qiskit`-Klasse

QAOA Zugriff via **Use Cases – SEQUOIA (fraunhofer.de)** [UC 3] bietet, einen einfachen Zugang zu Quantenalgorithmen erlaubt, jedoch nur mäßig effizient für konkrete Probleme implementiert ist. Dies liegt insbesondere an vielen Pre- und Postprocessing-Schritten, welche zwar das Ergebnis verbessern, jedoch auch sehr rechenintensiv sind. Ein fairer Vergleich verschiedener (klassischer und Quanten-) Algorithmen und verlässliche Aussagen zur Effizienz und Skalierung von QAOA sind somit nicht möglich. Genau andersherum verhält es sich mit eigenen low-level Implementierungen von QAOA, welche beispielweise basierend auf `qiskit.opflow` und `qiskit.quantum_info` realisiert werden können. Hier ist eine performante Umsetzung des Quantenalgorithmus angepasst auf den Use Case möglich und man hat volle Kontrolle über die verschiedenen Schritte bis zum Endresultat. Eine solche Umsetzung erfordert jedoch umfangreiches Expertenwissen sowohl in der Theorie des Quantenalgorithmus als auch in der Implementierung in Python und Qiskit. Als guter Mittelweg hat sich die `qiskit`-Klasse `QAOA-Ansatz` [UC 4] erwiesen, welche die QAOA-Schaltkreise generiert, während die restliche Funktionalität des Algorithmus vom Nutzer bereitgestellt wird. Eine detaillierte Dokumentation dieses Wissens zu QAOA und dem Vorgehen und der Implementierung ist in [UC5] verfügbar. In [UC 8] ist darüber hinaus die Adaption von QAOA zum verbesserten CVaR-QAOA implementiert und analysiert.

Aufbauend auf diesen Arbeiten ist die Ausführung der QAOA-Schaltkreise auf realen Quantencomputern (insbesondere dem IBM Quantum System One Ehningen) angegangen worden. Als erste große Herausforderung stand hier die Transpilation der Schaltkreise auf die IBMQ Systeme an. Es stellte sich heraus, dass die naive Verwendung des `Qiskit`-Transpiler keine zufriedenstellenden Ergebnisse erzielen kann. Ein genaues Wissen über die verschiedenen Transpilationsschritte und ihre Steuerung (z.B. über die Parameter des Transpilers) als auch Techniken zur Mitigation von Hardwarefehlern (z.B. mittels Dynamical Decoupling und Auslesefehlermitigation, siehe auch Abschnitt 3.2.4 Fehlermitigations-Modul) sind zwingend notwendig. In [UC 6] wurden diese Techniken dokumentiert und demonstriert sowie Big Data Tools vorgestellt, mit denen die Resultate der Quantencomputer verarbeitet werden können. Es muss jedoch darauf hingewiesen werden, dass das gesamte Framework `Qiskit` und insbesondere auch der Transpiler sehr volatil sind und eine stete Integration bzw. Entfernung von Features stattfindet.

Durch all diese Vorarbeiten war es möglich eine systematische Untersuchung des Fokus-Anwendungsfall als Quanten-Anwendung in Form zahlreicher Experimente insbesondere auf dem IBM Quantum System One in Ehningen durchzuführen. Um den Rahmen nicht zu sprengen, werden an dieser Stelle nur einige wenige dieser Experimente in kurzen Worten dargestellt. Alle Experimente zusammen mit ausführlichen Details und Interpretationen sind in [UC 7] dokumentiert. Um den Einschränkungen der NISQ-Ära Rechnung zu tragen, wurden kleine Beispiele wie sie Abbildung 24 zeigt ausgeführt. Dafür werden acht Qubits

benötigt, was einer Größe entspricht, bei der man vom IBM Quantum System One in Ehningen (Quantenvolumen 64) noch aussagekräftige Resultate erhält (16 Qubit Beispiele sind in [UC7] dokumentiert). Als variationeller Algorithmus (siehe Abschnitt Variationelle Quantenalgorithmen) besteht QAOA aus zwei Teilen, einer wird auf dem Quantencomputer ausgeführt und einer auf dem klassischen Computer. Beim klassischen Anteil handelt es sich um eine Optimierung der freien Parameter in QAOA. Abhängig von dem QAOA-Hyperparameter  $p$  hat man dabei  $2p$  Parameter zu optimieren. In der Literatur wird oft berichtet, dass es sich hierbei um eine sehr herausfordernde Aufgabe für klassische Optimierer handelt. Dies konnte in den vorliegenden Experimenten bestätigt werden. So zeigt Abbildung 25 eine typische Optimierungslandschaft, durch welche ein Optimierer ausgehend von einem vom Nutzer festzulegenden Startpunkt navigieren muss. Dies gestaltet sich als schwierig, da die Landschaft stark oszillatorisch mit vielen lokalen Extrema ist. Auch ein Verschwinden der Gradienten (der sogenannte Effekt der Barren Plateaus) ist bereits auszumachen.

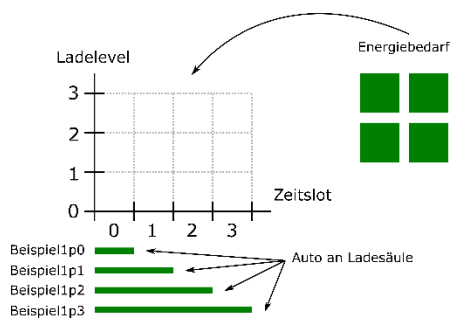


Abbildung 24: Vier Beispiele für Proof-of-Concept Modelle. Die Problemstellung besteht immer aus einem E-Auto, einer Ladesäule, vier verfügbare Zeitslots, vier mögliche Ladelevel und einer zu ladenden Energie. Die Beispiele unterscheiden sich lediglich in Ankunfts- und Abfahrtszeiten. Je kürzer das E-Auto an der Ladesäule ist, desto flacher werden später die QAOA-Schaltkreise.

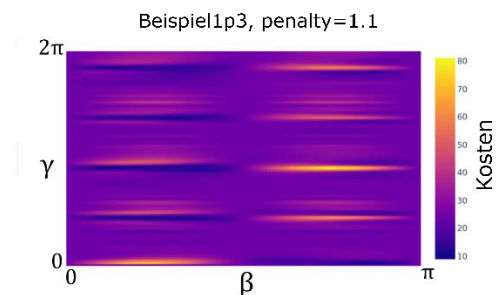


Abbildung 25: Optimierungslandschaft zu Beispiel1p3 (siehe Abbildung links) für QAOA mit  $p=1$ , d.h. mit zwei freien Parametern  $\beta$  und  $\gamma$ . Die Farbskala gibt den Wert der Kostenfunktion an. Ziel des klassischen Optimierungsteil in QAOA ist es Parameter zu finden, für welche die Kostenfunktion minimal ist.

Auch Abbildung 26 bestätigt nochmals die Schwierigkeit optimale Parameter zu erhalten. Hier sieht man das Ergebnis des klassischen Optimierers, wenn er von verschiedenen Startpunkten ausgeführt wird. Die vielen lokalen Extrema spiegeln sich hier in der großen Varianz der Ergebnisse wider.

Beispiel1p3

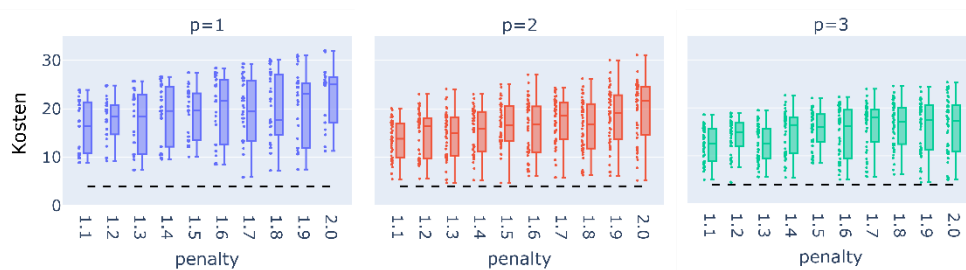


Abbildung 26: Ergebnis des klassischen Optimierers COBYLA für 50 zufällig gewählte Startpunkte  $\vec{\beta}_0 \in [0, \pi]^p$  und  $\vec{\gamma}_0 \in [0, 2\pi]^p$  mit einer exakten State-Vector-Simulation. Der Parameter penalty auf der x-Achse geht in der Transformation des Ausgangsproblems zu einem QUBO ein.

Durch die oben genannten Experimente konnten sehr gute bis optimale Wahlen für die Parameter in QAOA gefunden werden. Im nächsten Schritt wurden QAOA-Schaltkreise mit genau diesen Parametern auf dem IBM Quantum System One in Ehningen ausgeführt, um die Leistungsfähigkeit unter diesen bestmöglichen Bedingungen zu evaluieren. Es war zu erwarten, dass neben der Qubitanzahl vor allen Dingen die Tiefe der Schaltkreise eine entscheidende Rolle für die Qualität des Resultates spielt. Dies wurde im Design der Experimente zentral berücksichtigt und kann über die Verweildauer des E-Autos an der Ladesäule gesteuert werden (siehe Abbildung 24).

Durch viele Experimente kann folgendes bestätigt werden: Für flache Schaltkreise im Bereich 20 – 30 CNOT Gates (nach Transpilation) können gute Resultate vom IBM Quantum System One erhalten werden (Abbildung 27 links). Für mitteltiefe Schaltkreise (40 – 70 CNOT Gates) geht die Qualität merklich zurück, kann jedoch durch Fehlermitigationstechniken wieder entschieden verbessert werden (Abbildung 27 Mitte). Für noch tiefere Schaltkreise (über 80 CNOT Gates) ist ohne Fehlermitigationstechniken keine sinnvolle Lösung mehr zu erwarten und selbst wenn diese angewandt werden, ist die Varianz in der Qualität drastisch (Abbildung 27 rechts). An dieser Stelle muss unbedingt erwähnt werden, dass die Ergebnisqualität des IBM Quantum System One stark über die Zeit variiert (siehe Abbildung 28) und somit sämtliche Aussagen nur eine Momentaufnahme sind. Auch ist in Abbildung 28 zu sehen, dass die übliche Beurteilung eines Quantenschaltkreises vor allem aufgrund der CNOT Anzahl nicht immer eine valide Einschätzung der erzielten Qualität der Quantenlösung entspricht. Hier bedarf es noch weiterer Forschung.



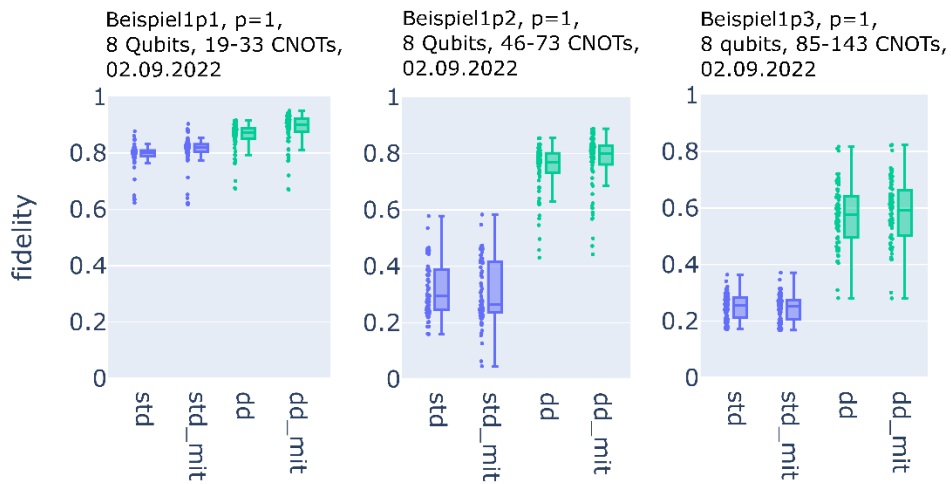


Abbildung 27: Qualität der vom IBM Quantum System One in Ehningen erhaltenen Lösung im Vergleich zu einer exakten Simulation (gemessen als fidelity, wobei 0 sehr schlecht und 1 sehr gut ist). Von links nach rechts sind drei verschiedene Beispiele, wobei das E-Auto links am kürzesten an der Ladesäule ist und rechts am längsten, siehe Abbildung 24. Die Tiefe der QAOA-Schaltkreise nimmt dadurch von links nach rechts zu. Es wurden 75 verschiedene Transpilationen durchgeführt, daher die variierende CNOT Gate Anzahl. Beschriftung der x-Achse: std=standard Transpilation mit Qiskit-Transpiler und `optimization_level=3`, dd=standard Transpilation und zusätzlich Dynamical Decoupling, xx\_mit=xx und zusätzlich Auslesefehlerkorrektur.

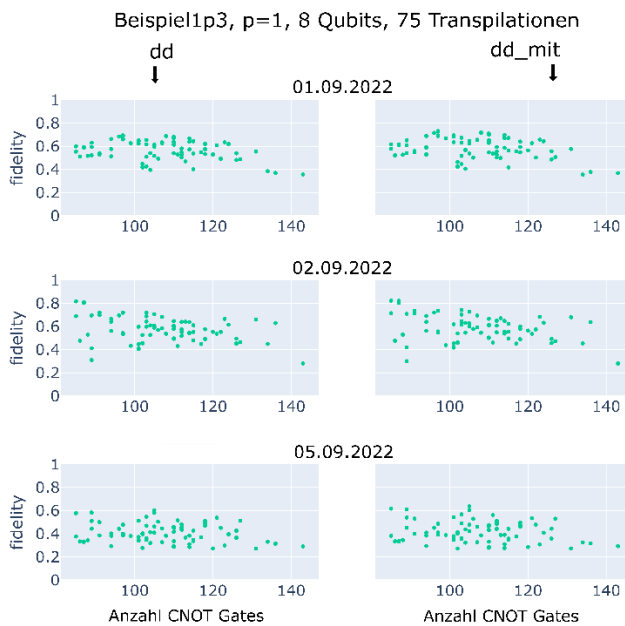


Abbildung 28: Qualität der vom IBM Quantum System One in Ehningen erhaltenen Lösung im Vergleich zu einer exakten Simulation des gleichen Beispiels an verschiedenen Tagen. Zur Bedeutung dd und dd\_mit und zum Begriff fidelity, siehe Abbildung 27.

Autor: Dr. Andreas Sturm (Fraunhofer IAO)

Kontakt: [andreas.sturm@iao.fraunhofer.de](mailto:andreas.sturm@iao.fraunhofer.de)

## 2.2.2 Fokusanwendung 2 »Quantenbasierte numerische Strömungssimulation«

Im zweiten Fokus-Anwendungsfall wird das Thema »Quantenbasierte numerische Strömungssimulation« behandelt. Ziel ist es, heutige Computational Fluid Dynamics (CFD) Rechnungen mit Quantencomputern zu beschleunigen und/oder in ihrer Approximationsgüte zu verbessern. Der hier betrachtete Use Case wurde in enger Zusammenarbeit zwischen dem Fraunhofer IAO, dem Höchstleistungsrechenzentrum HLRS der Universität Stuttgart und der Firma Simerics entwickelt.

Das Grundprinzip heutiger CFD-Rechnungen basiert auf dem Lösen (gekoppelter) Navier-Stokes-Gleichungen in drei Raum- und einer Zeitdimension. Um ein für die NISQ-Ära angemessenes Zielproblem zu definieren, wurde zusammen mit der Firma Simerics abgeklärt, initial die dreidimensionale Navier-Stokes-Gleichung unter bestimmten Randbedingungen zu lösen: Dabei wurde auf zwei Raum- und eine Zeitdimension fokussiert, wobei wir in longitudinale Richtung (in Strömungsrichtung) der Einfachheit halber zuerst Translationsinvarianz annehmen (d.h. effektiv nur zwei Dimensionen analysieren, eine Zeit- und eine Raumdimension). Dies ist schematisch für einen fixen Zeitpunkt in Abbildung 29 gezeigt.

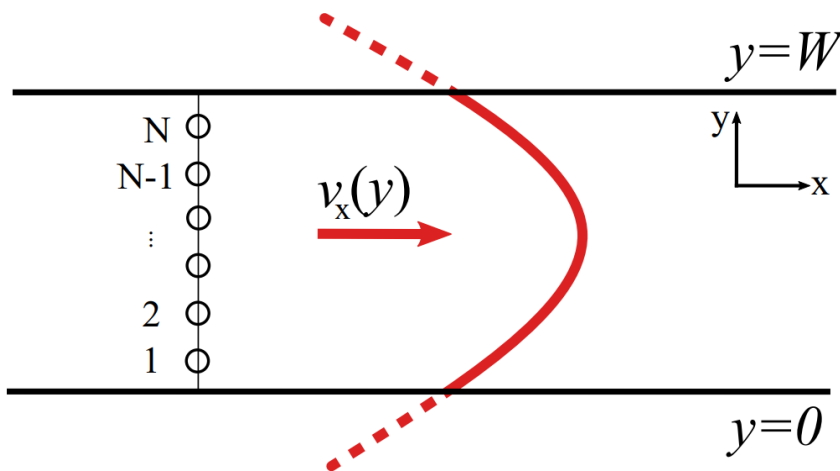


Abbildung 29: Schematisches Diskretisieren und Flussprofil einer zweidimensionalen Strömung zu einem gegebenen Zeitpunkt  $t = t_0$ .

Die Navier-Stokes Gleichung für den hydrodynamischen Fluss einer inkompressiblen Flüssigkeit mit Translationsinvarianz in die  $\vec{e}_x$ -Richtung und unter verschwindender Fluidgeschwindigkeit in  $\vec{e}_y$ -Richtung ist in zwei Raum- und einer Zeitdimension gegeben als

$$\frac{\partial u_x}{\partial t} = \nu \frac{\partial^2 u_x}{\partial y^2},$$

wobei  $u_{x_0}(t, y)$  die Strömungsgeschwindigkeit an der Stelle  $x = x_0$  als Funktion der Koordinaten  $y$  und  $t$  definiert. Die kinematische Viskosität ist durch  $\nu$  gegeben.

Diese spezielle Art der Gleichung ist auch als zweidimensionale Wärmeleitungsgleichung bekannt (eine Raumdimension, eine Zeitdimension), wobei die Viskosität als Temperaturleitfähigkeit interpretiert werden kann.

Eine populäre Methode zur Berechnung von approximativen Lösungen der Wärmeleitungsgleichung besteht darin die Differentialgleichung mittels der Finite Differenzen Methode im Ort und mit dem impliziten Eulerverfahren in der Zeit zu diskretisieren. Hierbei handelt es sich um ein numerisch sehr stabiles Verfahren, welches jedoch mit dem Preis kommt, dass in jedem Zeitschritt ein lineares Gleichungssystem gelöst werden muss. Genau dies soll im hiesigen Use Case durch passende (Quanten-) Algorithmen geschehen. Es geht also primär um das Lösen linearer Gleichungssysteme  $A\vec{x} = \vec{b}$ . Die detaillierte Herleitung dieser Gleichung hinsichtlich der initialen Differentialgleichung (DGL) ist unserem [Demonstrator](#) zu entnehmen.

Insbesondere ist die hier diskutierte DGL in Ihrer diskretisierten Form durch eine tri-diagonale Matrix beschrieben, das heißt eine Matrix in der lediglich die Diagonale und die Nebendiagonalen besetzt sind.

**Das Lösen von linearen Gleichungssystemen mittels des »reinen« Quantenalgorithmus HHL [UC 10]** hat sich aufgrund der zu hohen Schaltkreistiefe als nicht zielführend auf aktueller Quantenhardware erwiesen (vgl. Schaltkreistiefe und Qubit-Skalierung in Abbildung 30). Daraufhin wurde der Variational Quantum Linear Solver [UC 11] (VQLS) als variationelle, hybride Alternative in einer eigenen Implementierung untersucht. Dies geschah sowohl in Simulationen mit und ohne Noise-Modell als auch auf realer Quantenhardware (insbesondere auf dem IBM Q System One in Ehningen).

VQLS:			HHL:		
Matrix	Qubits	Tiefe	Matrix	Qubits	Tiefe
4x4	3	11	4x4	6	1096
8x8	4	15	8x8	8	<b>36962</b>

Abbildung 30: Schaltkreistiefe und Qubitanzahl des VQLS- vs. HHL-Algorithmus in Abhängigkeit der Größe des zu lösenden linearen Gleichungssystems.

Eine zentrale Rolle im VQLS-Algorithmus spielt die Wahl der Kostenfunktion. Anhand deren Wert bemisst sich die Güte des iterativen Lösungsvorschlags im hybriden quanten-

klassischen Optimierungsprozess. Eine hohe Kostenfunktion bedeutet eine schlechte Approximation der Lösung des linearen Gleichungssystems und umgekehrt.

Im VQLS Algorithmus gibt es zwei Alternativen die Kostenfunktion zu definieren, welche konkret Abbildung 31 zu entnehmen sind: Die globale Kostenfunktion  $C_G$  und Ihr lokales Pendant  $C_L$ .

Während die globale Kostenfunktion  $C_G$  keine lokalen Minima in der Optimierungslandschaft aufweist, zeigt sie jedoch sogenannte Barren Plateaus bei skalierender Qubitanzahl. Die lokale Kostenfunktion  $C_L$  hingegen umgeht jenen Effekt und geht ebenfalls mit deutlich geringeren Schaltkreistiefen einher (insbesondere mit nur zwei Hadamard Tests pro Evaluierung). Sie scheint daher in der NISQ Ära präferiert, kommt jedoch mit einer deutlichen Herausforderung: Man kann im klassischen Optimierungsschritt leicht in lokalen Minima „hängenbleiben“. Dies definiert eine besondere Herausforderung an die zu wählenden klassischen Optimierer.

Der Übersichtlichkeit halber haben wir in Abbildung 31 eine performante Lösung eines linearen Gleichungssystems sowohl mit der lokalen als auch der globalen Kostenfunktion dargestellt. Man sieht, dass zwar in beiden Ansätzen die klassische Lösung reproduziert werden kann, die lokale Methode allerdings besser konvergiert. Dieses Verhalten ist insbesondere bei größeren Systemen deutlich zu erkennen, kommt aber mit dem Problem lokaler Minima, bei denen eine konvergierte Lösungen erreicht werden kann, welche jedoch einen völlig falschen Lösungsvektor darstellt.

$$C_G = 1 - \frac{\sum_{l,l'} c_l c_{l'}^* \langle 0|V^\dagger A_{l'}^\dagger U|0\rangle \langle 0|U^\dagger A_l V|0\rangle}{\sum_{l,l'} c_l c_{l'}^* \langle 0|V^\dagger A_{l'}^\dagger A_l V|0\rangle}$$

$$C_L = \frac{1}{2} - \frac{1}{2n} \frac{\sum_{j=0}^{n-1} \sum_{l,l'} c_l c_{l'}^* \langle 0|V^\dagger A_{l'}^\dagger U Z_j U^\dagger A_l V|0\rangle}{\sum_{l,l'} c_l c_{l'}^* \langle 0|V^\dagger A_{l'}^\dagger A_l V|0\rangle}$$

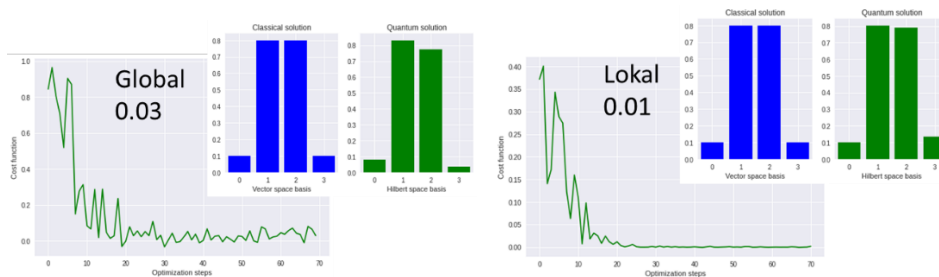


Abbildung 31: Iteratives Lösen des linearen Gleichungssystems  $A\vec{x} = \vec{b}$  mittels der globalen Kostenfunktion  $C_G$  und lokalen Kostenfunktion  $C_L$  mittels QASM Simulator. Die Definition der einzelnen Variablen und Operatoren sind [QCFD\_2] zu entnehmen.

Bei systematischer Analyse der gängigen klassischen Qiskit-Optimierer hat sich herausgestellt, dass die Performanz dieser stark vom Noise-Modell oder dem (echten) Rauschen des IBM Quantum System One abhängig ist. Ohne Fehlerquellen kristallisierte sich bei der Analyse der COBYLA als bester Optimierer heraus, während im Beisein von Fehlerquellen der SPSA Optimierer das beste Ergebnis lieferte.

In Abbildung sind zwei Beispiele des VQLS-Algorithmus auf dem IBM Quantum System One Ehningen mit lokaler Kostenfunktion und unterschiedlichen klassischen Optimierern dargestellt, welche ein schlechtes Konvergenzverhalten zeigen, während Abbildung eine erfolgreiche Minimierung der Kostenfunktion zeigt, allerdings erst nach Kalibrierung.

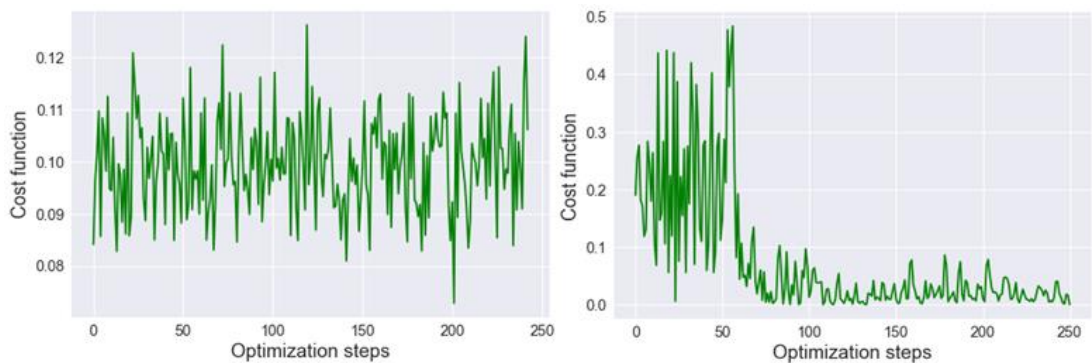


Abbildung 32: Konvergenzverhalten des klassischen Optimierers SPSA im VQLS-Algorithmus auf dem IBM Quantum System One Ehningen mit lokaler Kostenfunktion ohne Kalibrierung (links) und mit Kalibrierung (rechts). Nach Kalibrierung ist eine deutliche Konvergenz zu erkennen.

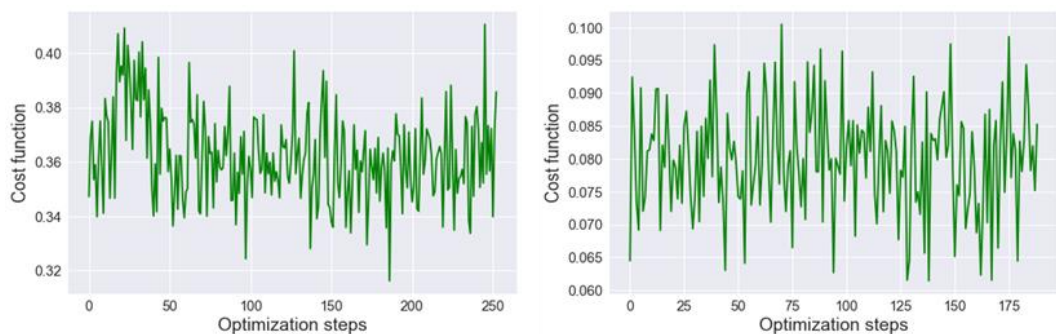


Abbildung 33: Konvergenzverhalten der klassischen Optimierer »Conjugate Gradient« (links) und »Truncated Newton« (rechts) im VQLS-Algorithmus auf dem IBM Quantum System One Ehningen mit lokaler Kostenfunktion. Es ist keine Konvergenz zu erkennen.

Wie erwähnt ist neben der Konvergenz des klassischen Optimierers auch die zugehörige Qualität der Lösung im VQLS-Algorithmus von besonderem Interesse, da gerade auch lokale Minima in der lokalen Kostenfunktion approximiert werden können. Dies wurde in zahlreichen Beispielen (für verschiedene Matrizen und Vektoren) auf dem IBM Quantum System One untersucht, wobei aufgrund der Beschränkungen der NISQ-Ära nur kleine lineare Systeme betrachtet werden konnten. Abbildung 30 stellt das Ergebnis des VQLS-Algorithmus für den Fall einer 4 x 4 Identitätsmatrix dar. Hier ist eine gute Qualität der Lösung zu erkennen. Wird die Schwierigkeit auf eine 8 x 8 Identitätsmatrix erhöht, ist

bereits eine Verschlechterung des Resultates zu sehen (siehe Abbildung 35). Für den Fall einer Tridiagonalmatrix, wie sie oft bei Diskretisierungen von partiellen Differentialgleichungen auftritt (siehe obige Diskussion), ist in Abbildung 36 zu erkennen, dass die Lösung des VQLS-Algorithmus selbst für 4 x 4 Systeme nur noch von mäßiger Qualität ist.

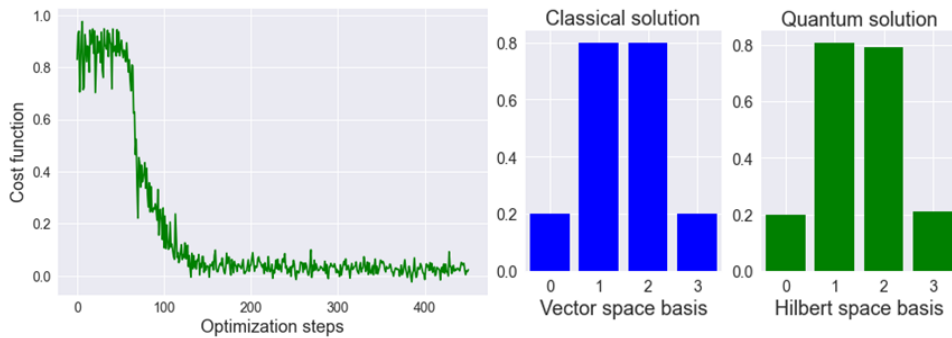


Abbildung 32: Konvergenzkurve (links) und Resultat (rechts) des VQLS-Algorithmus auf dem IBM Quantum System One in Ehningen. Als Optimierer kam SPSA zum Einsatz, als Matrix  $A$  die 4 x 4 Identitätsmatrix und als rechte Seite  $\vec{b}$  der Vektor  $[0.2, 0.8, 0.8, 0.2]$ . In der Mitte ist die exakte Lösung dargestellt.

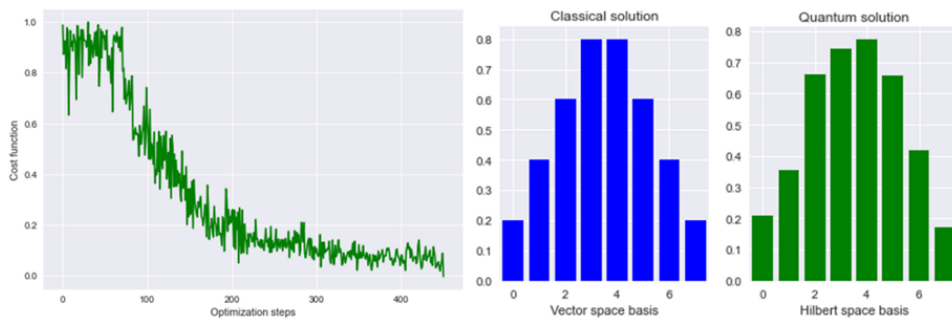


Abbildung 33: Situation wie in Abbildung 30, aber als Matrix  $A$  kam die 8 x 8 Identitätsmatrix zum Einsatz und als rechte Seite  $\vec{b}$  der Vektor  $[0.2, 0.4, 0.6, 0.8, 0.8, 0.6, 0.4, 0.2]$ .

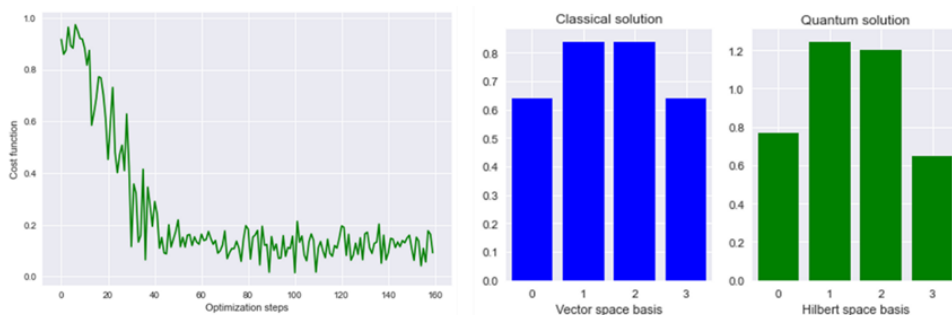


Abbildung 34: Situation wie in Abbildung 31 aber als Matrix  $A$  wurde eine tridiagonale Matrix verwendet sowie als rechte Seite  $\vec{b}$  der Vektor  $[1, 1, 1, 1]$ .

Um die Performanz des Algorithmus hinsichtlich der Lösung auch komplexerer Strömungsprobleme zu erhöhen, wurden im letzten Schritt in unsere VQLS-Algorithmen, die in AP3 entwickelten Fehlermitigations-Strategien inkludiert. Jene Lösung, sowie unser finaler Demonstrator wurde als interaktives Jupyter Notebook auf der KQCBW Recheninfrastruktur deployt und ist dort seit Januar 2023 der breiten Öffentlichkeit open source zugänglich.

Link: <https://websites.fraunhofer.de/sequoia/Use Cases/>

Abschließend sei anzumerken, dass noch weitere Forschung für den Dateninput (Encoding des  $\vec{b}$  Vektors und Pauli-Zerlegung der Matrix  $A$ ) sowie den Datenoutput (Auslesen des Lösungsvektors) notwendig ist, welche sonst bei Skalierung der Systemgröße mögliche Quantenvorteile wieder revidieren könnten.

Autoren: Dr. Christian Tutschku (Fraunhofer IAO), Dr. Bharadwaj Chowdary Mummaneni (Fraunhofer IAO)

Kontakt: [christian.tutschku@iao.fraunhofer.de](mailto:christian.tutschku@iao.fraunhofer.de),  
[bharadwaj.chowdary.mummaneni@iao.fraunhofer.de](mailto:bharadwaj.chowdary.mummaneni@iao.fraunhofer.de)

### 2.2.3 Fokusanwendung 3 »Optimierung einer Auftragssequenzierung in der diskreten Fertigung«

Der dritte Use Case »Optimierung einer Auftragssequenzierung in der diskreten Fertigung« behandelt die Verteilung von Produktionsaufträgen auf unterschiedlichen Produktionsressourcen und wurde gemeinsam vom Fraunhofer IPA mit der KUMAVISION AG erarbeitet. Diese Problemstellung existiert in nahezu allen fertigenden Branchen jedoch variiert sie meist im Detail bei den abzubildenden Einflussfaktoren (z.B. Materialverfügbarkeit, Mitarbeiterfähigkeiten, Anlagenkapazitäten etc.), Randbedingungen oder zu optimierenden Zielgrößen.

Eine in der Literatur weit verbreitete Definition dieser Problemart ist das sogenannte Job Shop Scheduling Problem (JSP). Abzuarbeiten sind hierbei eine Reihe von Produktionsaufträgen (Jobs), bei denen jeder Job aus beliebig vielen Fertigungsschritten (Operationen) bestehen kann, die in einer vorgegebenen Reihenfolge abgearbeitet werden müssen. Jeder Fertigungsschritt muss auf einer spezifischen Anlage (Maschine) durchgeführt werden. Die Anlagen unterliegen dabei einer Reihe von Einschränkungen. So müssen die vorangegangenen Fertigungsschritte eines jeden Jobs vollendet sein, bevor der aktuelle Fertigungsschritt bearbeitet werden kann. Eine Anlage kann außerdem zu

jedem Zeitpunkt nur einen Fertigungsschritt bearbeiten. Fertigungsaufträge können nicht pausiert werden. Ziel des klassischen JSP ist die Minimierung der Gesamtproduktionsdauer (englisch Makespan).

Abbildung 35 zeigt eine beispielhafte (nicht-optimale) Lösung eines JSP auf drei Maschinen. Die Produktionsschritte, die zu ein und demselben Job gehören, sind hier farblich zusammengefasst. Die Anordnung führt zu einer Gesamtproduktionsdauer von 5 Zeiteinheiten.

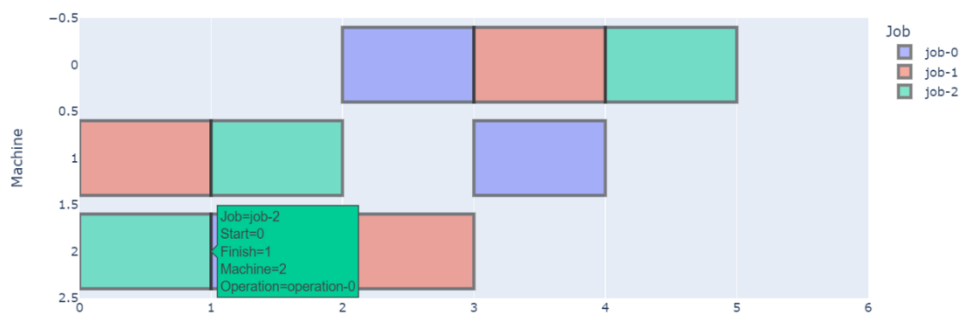


Abbildung 35: Exemplarische Lösung für ein Job Shop Scheduling Problem

Im Rahmen dieser Fallstudie wurde das JSP in Form des sogenannten Time-Indexed-JSP-QUBO [JSP1] (QUBO, englisch für quadratic unconstrained binary optimization problem) implementiert. Besonderer Fokus lag auf der Bestimmung optimaler Parameter (englisch penalty parameter), welche garantieren, dass die Randbedingungen des Problems eingehalten werden. Außerdem wurden Heuristiken implementiert, die die benötigten Binärvariablen und den damit verbundenen Lösungsraum reduzieren (englisch Variable Pruning). Beispielsweise wurden absurde Startzeiten für Fertigungsschritte aus dem Lösungsraum entfernt. Dies führte für eine Minimalinstanz des QUBOs zu einer Formulierung, die 9-Qubits benötigt, um das Problem zu beschreiben. Mithilfe der Qiskit Implementierung des QAOA-Algorithmus konnten diese minimalen JSP-Probleminstanzen auf dem IBM Q System One in Ehningen erfolgreich gelöst werden. Abbildung zeigt hierzu beispielhafte Ergebnisse. Die implementierte Lösung wurde in Form eines Demonstrators auf der [Projektwebsite](#) veröffentlicht.

Abschließend wurde versucht durch weitere Heuristiken den JSP-Use Case für größere Probleminstanzen mithilfe des QAOA-Algorithmus zu lösen. Hierzu wurde z.B. für Probleminstanzen, welche eine hohe Bearbeitungsdauer aufweisen, durch sogenanntes »Operation Duration Squashing« eine Variablenreduktion erzielt.



Aufbauend auf diesen bisherigen Resultaten wurde ein weiterer aus der Wissenschaft bekannter Ansatz für die Lösung von *Job-Shop Scheduling Problemen* hinzugezogen. Dieser unterscheidet sich in der Behandlung der zu erfüllenden Randbedingungen.

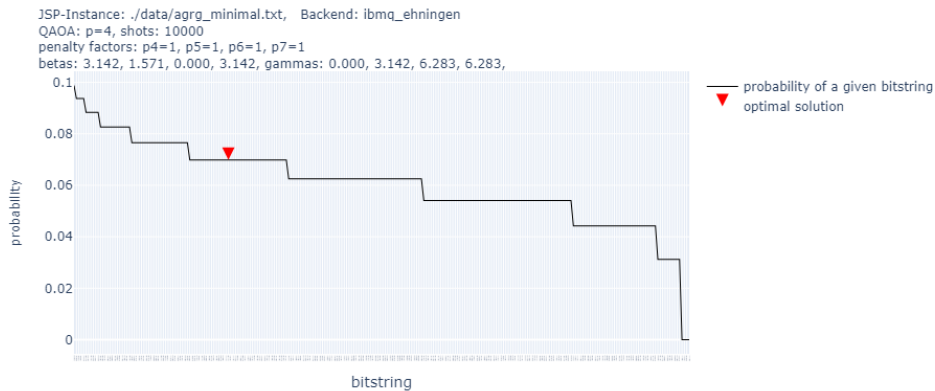


Abbildung 36: Exemplarische Wahrscheinlichkeitsverteilung der Lösungen bei Nutzung des QAOA-Ansatzes auf dem IBM System One (optimale Lösung rot markiert)

Es wurde oben bereits erwähnt, dass eine Schwierigkeit des Problems darin besteht, nicht zulässige Lösungen (z.B. wegen doppelter Maschinenbelegung oder falscher Job-Reihenfolge) sicher auszuschließen. Im oben beschriebenen Ansatz geschieht dies über die erwähnten »Strafparameter«. Diese bewirken, dass bei einer Optimierung unzulässige Konfigurationen aufgrund ihres hohen Kostenwertes nicht als Lösung ausgewählt werden. Dieser Ansatz nennt sich »soft constraints«, da bei zu niedriger Wahl der Strafparameter trotzdem unzulässige Lösungen auftreten können.

Der nun zu besprechende Ansatz der »hard constraints«<sup>9</sup> hingegen schließt nicht zulässige Lösungen sicher aus und kommt völlig ohne Strafparameter aus. Damit offenbart er bereits einen großen Vorteil: Die notwendige Optimierung der Strafparameter entfällt vollständig. Dafür ist es aber notwendig, den zulässigen Lösungsraum gut zu verstehen. Nur so kann man Schaltkreise entwerfen, die die Struktur des Lösungsraums respektieren, in dem Sinne, dass man ausgehend von zulässigen Lösungen wieder zu (anderen) zulässigen Lösungen kommt.

Zum Verständnis der Methode der »hard constraints« ist eine etwas abstraktere Sichtweise hilfreich: Bei der Beschreibung des Lösungsraums spielen Symmetrien eine wichtige Rolle. Etwa beim verwandten Problem des Handlungsreisenden können, ausgehend von einer beliebigen Startreihenfolge, alle möglichen Lösungen erreicht werden, indem hinreichend oft die Reihenfolge von nur zwei nacheinander besuchten Städten vertauscht wird. Damit hat man einfache Grundbausteine gefunden, die die

<sup>9</sup> Siehe Hadfield: Quantum Algorithms for Scientific Computing and Approximate Optimization 2018

Zulässigkeit von Lösungen erhalten und durch Rekombination alle möglich zulässigen Lösungen ergeben. Mathematisch gesprochen hat man die Symmetriegruppe des Lösungsraums gefunden und eine einfache Möglichkeit, diese zu explorieren. Implementierungstechnisch kann man nun auf Basis dieses Verständnis' Schaltkreise entwerfen, die diese Exploration auf dem Quantencomputer durchführen. Je besser das Verständnis der Symmetrie, desto effizienter kann dabei die Implementierung erfolgen.

Der Ansatz der »hard constraints« für die Quantenoptimierung wurde nun in einer freien, wissenschaftlichen Kooperation mit der Leibniz Universität Hannover im Rahmen der SEQUOIA Fallstudien entscheidend verbessert: Für verschiedene »Maschinen-Zuordnungsprobleme«, aber auch für das »Problem des Rundreisenden« wurden die vollständigen Symmetriegruppen gefunden. Dadurch wird die Menge der zulässigen Lösungen durch wenige Operatoren beschreibbar. Ein großer Vorteil, aber auch offene Fragen, ergeben sich nun aus der optimalen Implementierung der Symmetrien auf einem Quantencomputer.

Ein erster, einfacher Prototyp ist der Algorithmus »QGOA« (Quantum Group Optimization Algorithm), der genau die Symmetriegruppe beispielsweise eines »Maschinen-Zuordnungsproblems« durch mathematische Gruppentheorie implementiert. Der Vorteil hierbei ist, dass man mit vergleichsweise wenigen Operatoren alle benötigten Symmetrien implementieren kann. Dies hat direkte Auswirkung auf den Algorithmus, da man beweisen kann nur polynomiell viele klassische Parameter benötigt, um diesen Algorithmus nutzen zu können. Dies ist sehr wichtig, da es bei allgemeinen variationellen Algorithmen eine offene und auch unklare Frage ist, wie stark nun wirklich die klassische Optimierung zur Lösung des Problems beiträgt. Ein Nachteil von »QGOA« ist, dass dieser Algorithmus eine zwar polynomiell begrenzte, aber trotzdem große Menge an Quantenbits benötigt, die aus analytischer Sicht nicht notwendig wären. Dieses Problem des »klugen Encodings« ist weiterhin eine offene Forschungsfrage und noch nicht abschließend geklärt. Weil der Ansatz über Symmetriegruppen aber universell ist, kann man diesen Teil zukünftig jedoch nutzen, um in der Entwicklung von Quantenalgorithmen für restringierte kombinatorische Optimierungsprobleme Fortschritte zu erzielen.

Der Quanten-Algorithmus »QGOA« wurde auf dem IBM Quantum System One in Ehningen implementiert. Hierbei wurde auch ein effizienterer, gradientenbasierter Optimierungsalgorithmus zur Lösung der klassischen Optimierungsaufgabe eingesetzt. Es konnten wesentliche Punkte herausgearbeitet werden, wie beispielsweise, dass Messungen nicht die Eigenschaft haben, den Raum der zulässigen Lösungen im Allgemeinen invariant zu lassen. Dieser Punkt wird ebenfalls in zukünftige Forschung und Entwicklung eingebunden werden.

Autoren: Dr. Florian Knäble (Fraunhofer IAO), Gereon Koßmann (Fraunhofer IAO), Arthur Grigorjan (Fraunhofer IPA)

Kontakt: [florian.knaeble@iao.fraunhofer.de](mailto:florian.knaeble@iao.fraunhofer.de), [gereon.kossmann@iao.fraunhofer.de](mailto:gereon.kossmann@iao.fraunhofer.de),  
[arthur.grigorjan@ipa.fraunhofer.de](mailto:arthur.grigorjan@ipa.fraunhofer.de)

## 2.2.4 Fokusanwendung 4 »Optimale Auslegung von Schnittmustern in der Blechfertigung«

Im Rahmen des Projektes »SEQUOIA« wurde ein Use Case untersucht, in dem die optimale Auslegung von Schnittmustern in der Blechfertigung erarbeitet wurde. Der Use Case wurde von der Trumpf Gruppe eingebracht und vom Fraunhofer IPA untersucht.

In der Blechfertigung werden Teile hergestellt, die durch Schneiden mit Laser- oder Stanzmaschinen aus großen Blechtafeln herausgelöst werden. Die Form der zu produzierenden Teile kann dabei sehr komplex sein. Weiterhin können verschiedene Teile aus dem gleichen Blech gestanzt werden. Um die Materialausnutzung pro Blechtafel zu maximieren, ist es wünschenswert, die zu schneidenden Teile so anzuordnen, dass die ungenutzte Fläche dabei minimal ist. Hierbei handelt es sich um ein Optimierungsproblem, bei dem die derzeitigen Lösungen rechenaufwändig sind, sodass oft kein optimales Ergebnis erzielt werden kann und manuelle Nachbereitung durch einen Experten nötig ist. Die größte Schwierigkeit ergibt sich aus der freien Formgebung der Teile, wodurch es u.a. nötig ist, Teile zu rotieren oder ineinander zu schachteln, um eine optimale Platzausnutzung zu erzielen.

Der Use Case lässt sich mathematisch als zweidimensionales Zuschnittproblem (*Strip Packing Problem*) interpretieren. Hierbei wird eine vorgegebene Anzahl von  $N$  Teilen auf eine rechteckige Fläche mit fester Höhe  $H$  und variabler Länge  $L$  verteilt. Ziel ist es, die Teile so anzuordnen, dass  $L$  und somit auch die benötigte Gesamtfläche minimal sind. Um ein sinnvolles Ergebnis zu erhalten, müssen alle Teile so verteilt sein, dass sie komplett innerhalb der Fläche sind und dabei nicht überlappen. Dies ist beispielhaft in Abbildung illustriert. Die anzuordnenden Teile sind in der linken, eine optimale Konfiguration der Teile in der rechten Grafik zu sehen.

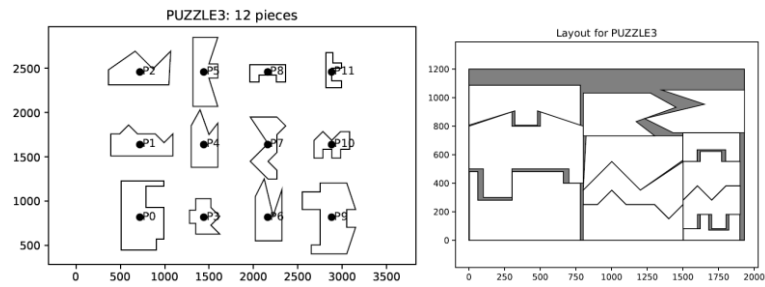


Abbildung 37: Beispielhafte Darstellung typischer Teile, die in unseren Berechnungen platziert werden (links). Resultat eines erfolgreichen Packvorgangs (rechts).

Neben der unmittelbaren Bedeutung für den vorliegenden Use Case ist dieses Problem darüber hinaus auch von erheblicher wirtschaftlicher Bedeutung. Durch die Lösung des Problems kann die verarbeitende Industrie ihre Kosten für Materialien wie Stoff, Leder, Karton, Holz, Kunststoff, Glas, Keramik oder Metall minimieren. Diese werden für die Herstellung von Einzelteilen verwendet, aus denen sich Endprodukte zusammensetzen, wie z. B. Kleidung, Autos, Schiffe, Elektrogeräte, Maschinen und Verpackungen. Eine Fertigung mit minimalem Verschnitt ist nicht nur ökonomisch, sondern auch ökologisch wünschenswert.

Zur Lösung des zweidimensionalen Zuschnittproblems werden üblicherweise heuristische Verfahren verwendet. Die meisten in der Literatur betrachteten Verfahren wählen eine feste Orientierung der Teile. Dies führt nur in Spezialfällen zu optimalen Lösungen. Eine besondere Anforderung an die Lösung des Use Cases war es deshalb, dass die Teile nicht nur frei positioniert, sondern auch rotiert werden können. Für die Lösung des Use Cases wurde ein hybrider, quanten-klassischer Algorithmus entwickelt. Hierbei wird das Problem in kleinere Unterprobleme zerlegt. In einem ersten Schritt werden Teile in Gruppen unterteilt. Die Gruppen werden so gewählt, dass Teile in eine möglichst kleine rechteckige Fläche passen. Die optimale Anordnung der Teile in diesen Rechtecken kann durch das Lösen eines Traveling Salesperson Problems<sup>10</sup> gefunden werden. Dies kann z.B. mit Hilfe des QAOA-Algorithmus (siehe QAOA) gemacht werden. Sind alle Teile überlappungsfrei verteilt, werden die kleinen Rechtecke auf der Gesamtfläche so angeordnet, dass sie möglichst wenig Platz einnehmen. Dies kann effizient mit klassischen Algorithmen gemacht werden. Der Algorithmus ist in der Infobox 16 genauer beschrieben.

---

<sup>10</sup> Das Traveling Salesperson Problem wird in den folgenden Use Cases 2.2.5 und 2.2.6 noch eingehender betrachtet und erläutert.

Im Rahmen des Projekts SEQUOIA wurde der Ansatz an acht verschiedenen Teilekonfigurationen getestet. Die Beispiele sind aus der Literatur entnommen [UC 17] und entsprechen gängigen Testinstanzen die zum Benchmarking der Leistung von Pack-Algorithmen verwendet werden. Um die Probleminstanzen für NISQ Hardware zugänglich zu machen, wurde der Rotationsfreiheitsgrad auf zulässige Winkel (vielfache von  $45^\circ$ ) beschränkt. Der QAOA-Schaltkreis besitzt  $p = 5$  Wiederholungen und wird mit dem COBYLA-Optimierer optimiert. Für die Lösung der QUBOS wurden 16 Qubits verwendet. Dies erlaubt eine maximale Anzahl von 4 Städten pro TSP. Der Schaltkreis wurde zunächst mit qiskits qasm-Simulator voroptimiert. Die so trainierten Schaltkreise wurden dann auf dem IBM Quantum System One in Ehningen ausgeführt. Die Anzahl der Shots wurde auf 1000 pro Schaltkreis festgelegt. Für jeden Schaltkreis wurde der Zustand mit der höchsten gemessenen Anzahl als Lösung ausgewählt. Dieses Vorgehen führt dazu, dass auch Lösungen, die die Randbedingungen des Optimierungsproblems verletzen ausgewählt werden können. In einem Nachbearbeitungsschritt wurde deshalb sichergestellt, dass der kodierte Pfad jede Stadt genau einmal besucht. Exemplarische Ergebnisse sind in Abbildung zu sehen.

### Infobox 15: Algorithmus zum Lösen des zweidimensionalen Zuschnittproblems

Analyse der geometrischen Kompatibilität zwischen Teilen. Für jedes Paar aus anzuordnenden Teilen wird eine optimale Anordnung (Rotation, Positionierung) ermittelt.

Bildung von Gruppen geometrisch kompatibler Teile. Teile, die sich im vorigen Schritt als besonders kompatibel herausgestellt haben, werden in Gruppen eingeteilt.

Bildung kompakter Rechtecke aus jedem Cluster. Wir ordnen die Teile jeder Gruppe räumlich nacheinander so an, dass die ungenutzte Fläche innerhalb eines rechteckigen Begrenzungsrahmen minimal ist. Für einen Cluster mit  $n$  Teilen gibt es zunächst  $n!$  mögliche Reihenfolgen. Bei einem Cluster mit 4 Teilen gibt es beispielsweise 24 solcher Sequenzen. Bei einem Cluster mit 5 Teilen wächst die Anzahl der Sequenzen auf 120 und bei 6 Teilen auf 720... Die Anordnung der Teile gemäß einer gegebenen Sequenz ist sehr rechenaufwändig, daher wollen wir dies nicht für jede mögliche Sequenz tun. Wir brauchen also ein einfaches Verfahren, um eine gute Reihenfolge direkt zu bestimmen. Wir können beobachten, dass die gesamte verschwendete Fläche ungefähr durch die Summe der verschwendeten Flächen zwischen jedem Paar von Teilen entlang der Sequenz begrenzt wird. Auf der Grundlage dieser Beobachtung können wir die Platzierungsreihenfolge wählen, die den Gesamtverschnitt minimiert. Sie entspricht der Summe  $\sum_{k=1}^{n-1} d(P_{i_k}, P_{i_{k+1}})$ . Die Suche nach dieser besten »Reihenfolge« für das Packen einer Menge von Teilen ist mathematisch äquivalent zur Lösung des so genannten *Traveling Salesperson Problems* (TSP). Dies kann mit dem Quantencomputer z.B. mit dem QAOA-Algorithmus erfolgen.

Optimale Anordnung der Rechtecke. Für jede Gruppe von Teilen lässt sich das ursprüngliche Packproblem auf ein einfacheres Problem reduzieren, bei dem nur Rechtecke aus den Clustern der Partition auf der Fläche platziert werden müssen. Das Problem, Rechtecke in einen rechteckigen Behälter zu packen, wird als Rechteckpacken bezeichnet. Hierbei handelt es sich um ein NP-schweres Optimierungsproblem, das heuristisch z.B. mithilfe des MAXRECTS-Algorithmus [UC 15] gelöst werden kann.

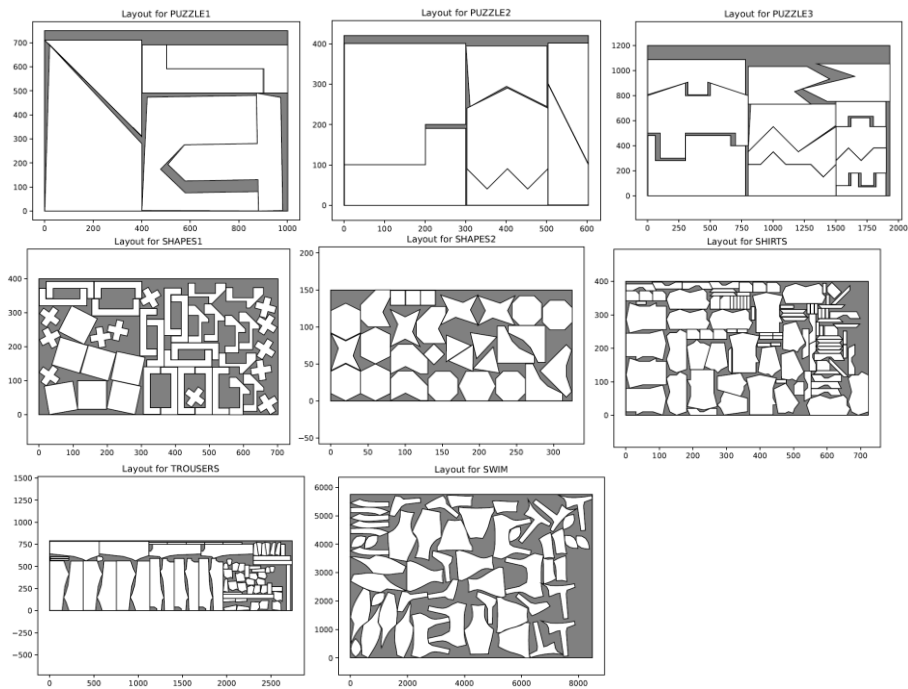


Abbildung 38: Exemplarische Resultate des quanten-hybriden Packvorgangs verschiedener Beispiele – gerechnet auf dem IBM Quantum System One in Ehningen.

Quantencomputing wird erst seit kurzem für die Lösung von Packproblemen in Betracht gezogen. Es wurde zunächst zur Lösung des 1D-Bin-Packing-Problems, dann zur Lösung des 2D-Rechteck-Packproblems [UC 18] und mit unserem Ansatz nun auch zur Lösung des 2D-unregelmäßigen Packproblems eingesetzt. Diese Arbeit unterstreicht die aktuelle Bedeutung der Entwicklung und des Einsatzes von Dekompositionsmethoden bei der Lösung von realen Problemen größeren Umfangs auf NISQ-Hardware mit begrenzter Anzahl von Qubits. Auch wenn der aktuelle Ansatz vielversprechende Ergebnisse liefert, weist er einige Schwierigkeiten wie das Skalierungsverhalten auf größere Probleminstanzen und herausfordernde Optimierung des variationellen QAOA auf. Trotz der aufgetretenen Schwierigkeiten zeigt dieser Use Case, dass die Entwicklung quantenklassischer Hybridalgorithmen ein zielführender Ansatz ist, der die Entwicklung neuer Konzepte und Methoden zur Lösung rechenintensiver Probleme ermöglicht.

Autoren: Dr. Paul-Amaury Matt (Fraunhofer IPA), Dr. Marco Roth (Fraunhofer IPA)

Kontakt: [paul-amaury.matt@ipa.fraunhofer.de](mailto:paul-amaury.matt@ipa.fraunhofer.de), [marco.roth@ipa.fraunhofer.de](mailto:marco.roth@ipa.fraunhofer.de)

## 2.2.5 Fokusanwendung 5 »Szenario-basierte Routenplanung zur Absicherung von Fahrfunktionen«

Im fünften Fokus-Anwendungsfall »Szenario-basierte Routenplanung zur Absicherung von Fahrfunktionen« von der Universität Tübingen, dem FZI und der Firma Bertrandt wurde die Routenplanung für ein Testfahrzeug betrachtet. Dabei sollen Testszenarien für automatisierte Fahrfunktionen in der realen Umgebung möglichst effizient durchlaufen werden können. Beispielsweise soll die Erkennung und Interpretation von Ampelanlagen getestet werden. Zu diesem Zweck soll, vom Standpunkt des Testfahrzeugs aus, eine Strecke auf Basis von Kartendaten ermittelt werden, auf der in einer vorgegebenen Zeit möglichst viele Ampelanlagen bzw. bestimmte Varianten von Ampelanlagen passiert werden. Durch Berücksichtigung von Witterung, Verkehrsdichte und vorgegebene Fahrrichtungen steigt die Komplexität dieses Problems erheblich an. Es wurde deshalb zum einen in einer wirklichkeitsnäheren Variante als Orienteering Problem (OP) und auf den kombinatorischen Problemkern reduziert als Traveling Salesperson Problem (TSP) formuliert. Bei beiden handelt es sich also um kombinatorische Optimierungsprobleme [UC 19].

Ein erster Ansatz zur Lösung des eigentlichen Orienteering Problems auf dem Quantencomputer, der sich leider als wenig effizient erwiesen hat, besteht in der Formulierung als ILP und Überführung in ein QUBO, wobei für die nicht binären Variablen und Randbedingungen zusätzliche Variablen eingeführt werden müssen. Die automatische Abbildung des ILP durch Qiskit auf ein QuadraticProgram führt bereits für ein Minimalbeispiel zu großen Schaltkreisen, die auf dem Quantencomputer in Ehningen nicht sinnvoll ausgeführt werden können.

Alternativ können für viele Optimierungsprobleme Quantenschaltkreise »manuell« erstellt werden, die die konkreten Eigenschaften des Problems ausnutzen und somit kleinere/effizientere Schaltkreise zur Folge haben. Die grundlegende Idee ist hierbei Randbedingungen, die für das Optimierungsproblem gelten müssen, statt in die Kostenfunktion in den sogenannten Mixer des QAOA-Algorithmus (Quantum Alternating Operator Ansatz) einzubetten. Es ist bekannt, dass es eine solche optimierte Formulierung für das TSP gibt [UC 20]. Für das OP bleibt dies eine offene Fragestellung, da die im Gegensatz zu TSP nicht vorab bekannte Tourlänge eine entsprechende Formulierung stark erschwert.

Ein Algorithmus zur approximativen Lösung des TSP wurde mittels QAOA-Algorithmus auf dem Quantenrechner und Simulator umgesetzt und getestet. Dies führte zutage, dass selbst kleine Instanzen des Problems zwar lauffähig sind, aber hohe Fehlerraten auf dem Quantenrechner in Ehningen zur Folge haben. Die Implementierung wurde daher hinsichtlich der Schaltnetzgröße optimiert.



Die Schaubilder unten illustrieren beispielhaft die Ergebnisse auf dem Quantenrechner in Ehningen für die TSP-Implementierung mit drei Städten. Beide Bilder zeigen Histogramme der Messungen, nachdem der QAOA-Schaltkreis mit optimierten Parametern ausgeführt wurde. Das gewünschte Ergebnis ist zusätzlich rot markiert. Links sind die häufigsten Messergebnisse ohne weitere Nachbearbeitung dargestellt. Daraus ist zu erkennen, dass das gesuchte Ergebnis durch die verrauschten Berechnungen nicht klar als solches zu erkennen ist. Das wahrscheinlichste Messergebnis (höchster Balken) ist darüber hinaus eine ungültige TSP-Tour, verletzt also mindestens eine der Randbedingungen der Problemstellung. Trotz dem, dass eine QAOA-Variante implementiert wurde, die theoretisch ungültige Touren gar nicht im Zustandsraum zulässt, sind durch Gatter- und Auslesefehler solche Zustände gemessen worden.

Filtert man alle ungültigen Zustände aus den Messergebnissen heraus, ist das optimale Ergebnis als solches erkennbar, was im rechten Bild dargestellt ist. Für drei Städte gibt es nur 6 Möglichkeiten, sie in eine Reihe zu bringen – aus diesen Möglichkeiten wurde die optimale am häufigsten gemessen. Demnach muss für die sinnvolle Verwendung von Quantenrechnern großer Fokus daraufgelegt werden, Fehler zu reduzieren, zu mitigieren oder zu korrigieren.

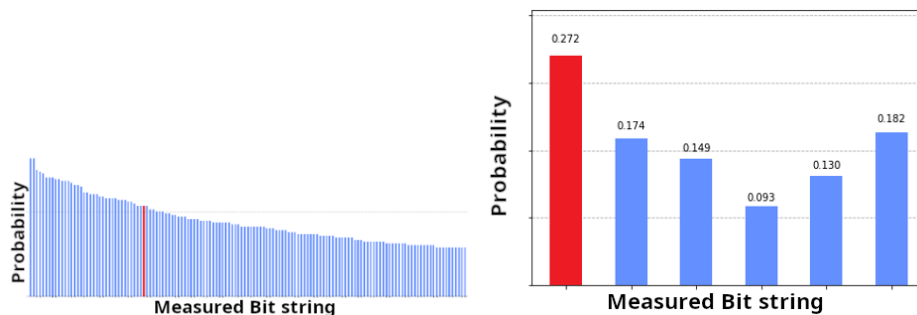


Abbildung 39: Verteilung der Häufigkeit der gemessenen Bitstrings direkt nach der Ausführung auf dem QC in Ehningen (links) und Verteilung der Häufigkeit der Strings, nachdem ungültige Touren herausgefiltert wurden

Die bisher existierenden, in ihrer Größe stark eingeschränkten Quantenrechner sollen möglichst früh Probleme lösen, deren Komplexität praktische Relevanz haben. Deshalb wurde nach Möglichkeiten gesucht, größere Probleminstanzen in kleinere aufzuteilen, um sie so auf kleinen Quantenrechnern berechenbar zu machen. Da durch schrumpfende individuelle Problemgrößen die auszuführenden Schaltkreise kleiner werden, kann so auch das entstandene Rauschen auf die Unterprobleme verteilt werden. Es wurden dabei Ansätze erarbeitet, die sich Elementen von dynamischer Programmierung und dem A\*-Algorithmus [UC 21] bedienen. Eine entscheidende Rolle spielt dabei ein Machine-Learning-Verfahren, mit dem die voraussichtliche Länge einer optimalen TSP-Tour abgeschätzt werden kann. Dies wird dazu benötigt, die Generierung der Unterprobleme zu

führen, da andernfalls exponentiell viele Unterprobleme generiert würden, deren Lösung zur Lösung des ursprünglichen Problems nicht beitragen, weshalb jeder Laufzeitvorteil verschwände.

In der Abbildung unten sind zwei verschiedene Lösungen für eine zufällig generierte TSP-Instanz zu sehen. Links ist die optimale Lösung dargestellt. Die Lösung rechts zeigt das Ergebnis des oben beschriebenen Algorithmus. Die generierten Teilprobleme bzw. deren Lösungen sind farblich voneinander abgegrenzt. Es wird daraus auch ersichtlich, dass es sich hierbei um ein Approximationsverfahren handelt, da Heuristiken und Schätzungen maßgeblich für die Güte des Ergebnisses verantwortlich sind. Gleichzeitig wird für die Lösung der Teilprobleme QAOA verwendet, was an sich nur approximative Ergebnisse liefert.

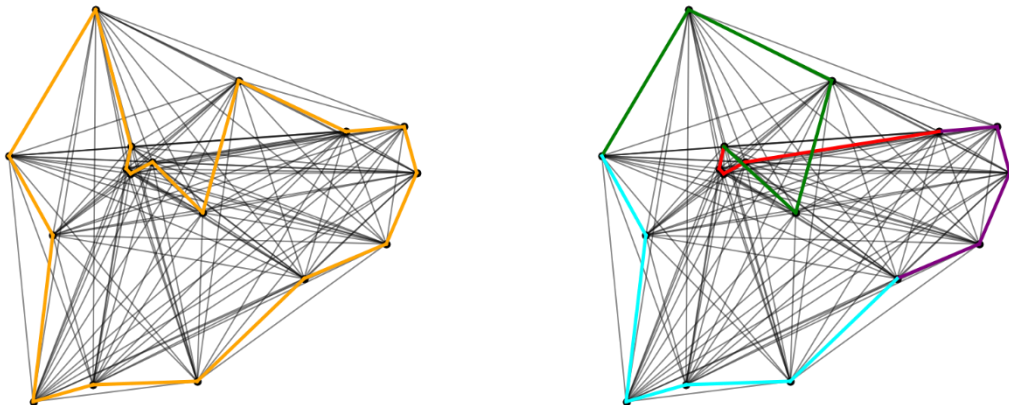


Abbildung 40: Darstellung einer optimalen TSP-Tour (links) und die approximative Lösung, die durch Problemaufteilung und Lösung auf dem Quantenrechner entstanden

Eine genauere Beschreibung der Funktionsweise dieses Algorithmus und dessen Implementierung sind in einem Demonstrator festgehalten worden.

Autor: Simon Garhofer (Eberhard-Karls-Universität Tübingen)

Kontakt: [simon.garhofer@uni-tuebingen.de](mailto:simon.garhofer@uni-tuebingen.de)

## 2.2.6 Fokusanwendung 6 »Routenplanung von LKW-Flotten im Supply-Chain-Management«

Im sechsten Fokus-Anwendungsfall »Routenplanung von LKW-Flotten im Supply-Chain-Management« vom Fraunhofer IAO mit der Firma Schwarz IT wurde ein Vehicle Routing Problem (VRP) modelliert, welches eine Verallgemeinerung des klassischen Traveling Salesperson Problems (TSP) darstellt.

Dabei geht es darum, eine große Zahl durchzuführender Lieferungen optimal auf eine kleinere Zahl von Lastwagen aufzuteilen. Sobald einem Lastwagen eine Auswahl von Lieferungen zugewiesen ist, entsteht ein kleineres Optimierungsproblem, welches man in erster Näherung als klassisches TSP auffassen kann. Genauer genommen handelt es sich um ein Pickup Delivery Problem (PDP). Dieses modifiziert das TSP in der Hinsicht, dass nicht mehr beliebige Abfolgen der Orte möglich sind, sondern gewisse Reihenfolgen eingehalten werden müssen (für jedes Paar von Abhol- und Lieferorten muss der Abholort vor dem Lieferort besucht werden).

In Abbildung ist ein VRP mit 9 durchzuführenden Lieferungen skizziert. Ein Pickup und Delivery Orte-Paar ist hier durch Gleichfarbigkeit und die Buchstaben P und D gekennzeichnet. Eine mögliche Partitionierung in drei Untermengen mit jeweils drei Lieferpaaren ist skizziert. Für eines der dadurch entstehenden PDPs ist eine optimale Lösung angegeben. Man beachte, dass innerhalb der gleichen Farbe immer P vor D durchlaufen wird. Alle Lastwagen starten vom selben Depot aus (schwarzer Punkt).

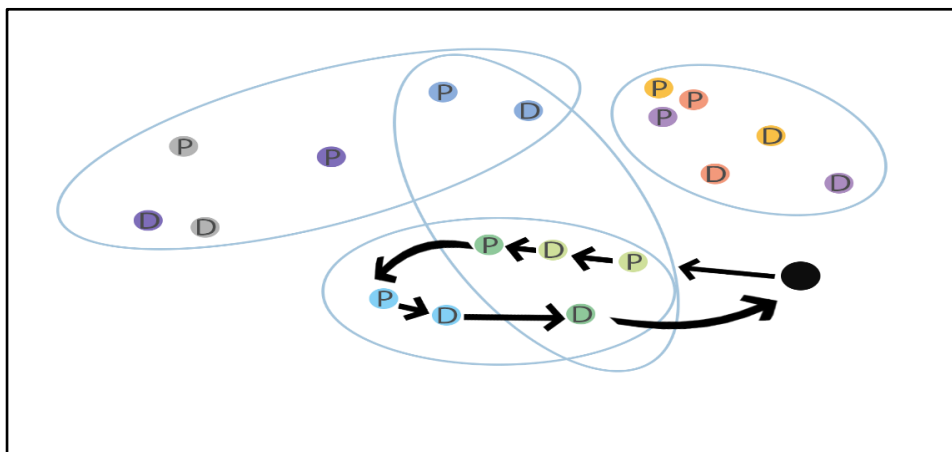


Abbildung 41: Ein Vehicle Routing Problem mit 9 Lieferungen als Set Partitioning Problem und die beispielhafte Lösung eines der drei dadurch entstehenden Pickup and Delivery Probleme.

In einem ersten Ansatz zur Lösung des gesamten VRP, also die gesamt-optimale Aufteilung der Lieferungen auf einzelne Lastwagen, wurde versucht das VRP als Set

Partitioning Problem (SPP) aufzufassen, welches sich durch eine Formulierung als QUBO ebenfalls für eine Lösung auf dem Quantencomputer anbietet. Dieser Ansatz hat sich leider nicht als sinnvoll erwiesen, da zur Formulierung der globalen Kostenfunktion die Lösung von sehr vielen kleinen PDPs nötig wäre und keine gute Heuristik zur Umgehung dieser Problematik gefunden wurde.

Anstelle des Set-Partitioning-Ansatzes wurde das VRP daher als Überlagerung einzelner PDPs dargestellt. Ein einzelnes PDP kann als spezielles Traveling-Salesperson-Problem (TSP) aufgefasst werden. Zur Einkodierung eines TSP mit  $n$  Orten sind  $n^2$  Qubits nötig, wenn man die Besuchsreihenfolge als Permutationsmatrix  $x_{i,j}$  repräsentiert. Für kleine Werte von  $n$  erhält man auch Problemgrößen, die für aktuelle NISQ-Architekturen noch handhabbar sind.

Im Folgenden soll kurz auf die oben schon angedeutete mathematische Formulierung und Einkodierung des TSP eingegangen werden, die das Problem erst für den Quantenrechner zugänglich macht. Die entsprechende Kostenfunktion  $f$ , die eine binäre Matrix  $x_{i,j}$  aufnimmt, und diese gemäß der zu fahrenden Strecken bewertet, stellt sich wie folgt dar:

$$f(x) = \sum_{i=1}^n x_{i,1} D_{n+1,i} + \sum_{t=1}^n \sum_{i,j=1}^n x_{i,t} x_{j,t+1} D_{i,j} + \sum_{i=1}^n x_{i,n} D_{i,n+1} + A \sum_{j=1}^n \left( 1 - \sum_{i=1}^n x_{i,j} \right)^2 + A \sum_{i=1}^n \left( 1 - \sum_{j=1}^n x_{i,j} \right)^2$$

$x_{i,j} = 1$  bedeutet hier, dass Tour  $i$  an Stelle  $j$  innerhalb der Abfertigungsreihenfolge steht.  $D_{i,j}$  stellt die (bekannten) Kosten für die Leerfahrt dar, um vom Ende der Tour  $i$  zum Anfang der Tour  $j$  zu gelangen.  $D_{n+1,i}$  entspricht der Entfernung vom Depot zum Beginn von Tour  $i$ . Der Parameter  $A$  ist ein Strafparameter, der sicherstellen soll, dass Binärmatrizen, die keine Permutationsmatrizen sind, nicht als Lösung in Frage kommen. Denn diese würden Konfigurationen entsprechen, die entweder Touren auslassen oder mehrere Touren gleichzeitig bedienen. Man beachte, dass die Kostenfunktion quadratisch in den  $x_{i,j}$  ist, damit haben wir das Problem als Quadratic Unconstrained Binary Optimization (QUBO) Problem formuliert, was uns die Nutzung des Quantenalgorithms QAOA zur Lösung ermöglicht.

In Abbildung 44 ist beispielhaft gezeigt wie eine Abfertigungsreihenfolge als Permutationsmatrix kodiert werden kann. Man beachte, dass hier aufgrund der Vereinfachung, dass Pickup und Delivery direkt aufeinander folgen müssen, eine Tour, die vorher aus zwei Orten bestand, nun auf einen Punkt abstrahiert wurde.

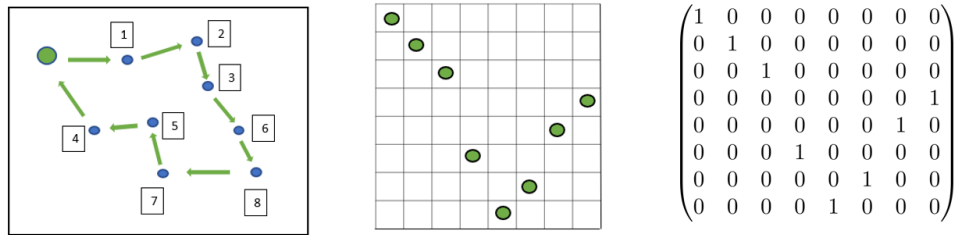


Abbildung 42: Eine mögliche Lösung eines Pickup and Delivery Problems eines LKW mit 8 Lieferungen und seine Enkodierung als Permutationsmatrix  $x_{i,j}$ .

Um ein VRP mit  $k$  Lastwagen und  $n$  Touren als QUBO formulieren zu können, kann man eine mögliche Aufteilung in Subtours über eine Partition einer Permutationsmatrix in  $k$  binäre Matrizen darstellen. Dafür sind dann  $k \cdot n^2$  Qubits notwendig. Die entsprechende Kostenfunktion lässt sich ähnlich wie oben im Falle von einem LKW aufstellen und sie ist immer noch quadratisch in den Matrixelementen. Abbildung 45 zeigt eine mögliche Aufteilung für zwei LKWs und die entsprechende Partition der Permutationsmatrix.

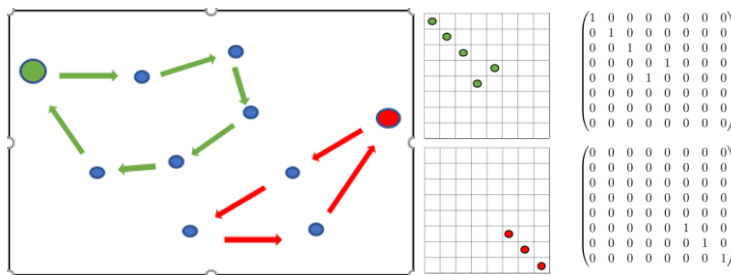


Abbildung 43: Eine mögliche Lösung eines Vehicle Routing Problems mit zwei LKWs und 8 Lieferungen und seine Enkodierung als zerlegte Permutationsmatrix.

In diesem Beispiel sind also bereits 128 Qubits notwendig. Ein auf dem IBM Quantum System One theoretisch noch darstellbarer Fall, sind etwa zwei Lastwagen und drei Touren mit 18 notwendigen Qubits. Im Folgenden wird eine Wahrscheinlichkeitsverteilung für einen LKW mit 3 Touren dargestellt, was 9 Qubits erfordert. Die entsprechenden Schaltkreise wurden mit Hilfe der Qiskit-Klasse QAOA-Ansatz erzeugt.

Die Wahrscheinlichkeitsverteilung der 1000 Shots ist sehr breit verteilt, das optimale Ergebnis rangiert zumindest an vorderer Stelle. Der Bitstring mit maximaler Wahrscheinlichkeit repräsentiert allerdings keine zulässige Lösung. Im Gegensatz zu Simulationen kann man aufgrund von Dekohärenzeffekten in echter Hardware auch mit passend gewähltem Strafparameter unzulässige Lösungen nicht immer vermeiden.

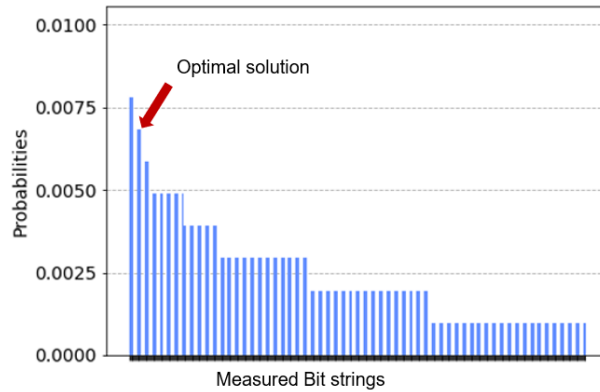


Abbildung 44: Wahrscheinlichkeitsverteilung der QAOA-Lösung eines Vehicle Routing Problems mit einem LKW und 3 Lieferungen gerechnet mit 1000 shots auf dem IBM Quantum System One in Ehningen. Die Optimale Lösung ist gekennzeichnet.

Abbildung 47 zeigt ein Simulationsergebnis mittels des QASM-Simulators. Man sieht, dass die Wahrscheinlichkeitsverteilung innerhalb einer Simulation viel gedrängter ist, da hier im Normalfall keine unzulässigen Lösungen auftreten. Auch wenn der favorisierte Bitspring mit 33% Wahrscheinlichkeit nur einer leicht suboptimalen Lösung entspricht, ragen die optimalen Lösungen mit 5% - 6% noch substantziell heraus.

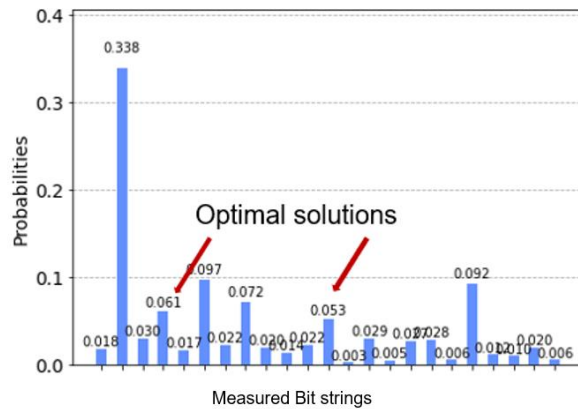


Abbildung 45: Wahrscheinlichkeitsverteilung der QAOA-Lösung eines Vehicle Routing Problems mit einem LKW und 4 Lieferungen gerechnet mit 1000 shots mit dem QASM-Simulator. Die beiden optimalen Lösungen sind gekennzeichnet.

Die Problemgrößen mit zwei LKWs, die 18 Qubits erfordern, lassen sich auf dem System One in Ehningen leider nicht mehr sinnvoll berechnen. Die hohe Qubitanzahl in Zusammenhang mit der starken Kopplung (jedes der 18 Qubits muss mit bis zu 12 anderen verschränkt werden) führt zu einer prohibitiv hohen Anzahl an CNOT Gattern, die im entsprechenden Schaltkreis bei den derzeitigen Fehlerraten zu so viel Rauschen führen, dass es von aktuellen Fehlermitigationstechniken nicht mehr beherrscht werden kann.

Die verwendeten Algorithmen sind auch in einem Demonstrator (siehe Projektwebsite) dargestellt. Hier wurden Beispielinstanzen von Vehicle Routing Problemen unterschiedlicher Komplexität in Python modelliert auf welche QAOA in unterschiedlicher Parametrisierung angewendet werden kann.

Autor: Dr. Florian Knäble

Kontakt: [florian.knaeble@iao.fraunhofer.de](mailto:florian.knaeble@iao.fraunhofer.de)

### 2.3 Expectation Management und Handlungsempfehlungen

In Kapitel 1.1 und Kapitel 2.1 wurde das sich rapide entwickelnde Ökosystem aus Hardware- und Softwareanbietern, Start-ups, Forschungspartnern und -projekten sowie interessierten Unternehmen diskutiert. Insbesondere wurden in Kapitel 2.1 konkrete industrielle Anwendungsfälle und deren Umsetzung mittels Quantentechnologie in der NISQ-Ära vorgestellt. Um jedoch allen Akteuren des Ökosystems eine gemeinsame und realistische Einschätzung der aktuell möglichen Lösungen via Quantencomputing zu vermitteln, einer Quantenhype vorzubeugen und etwaige Ansprüche an heutige Quantenlösungen realistisch einzuschätzen, soll an dieser Stelle noch einmal konkret auf den Stand der derzeitigen Forschung eingegangen werden.

Wie in Kapitel 2.1 beschrieben, sind die heute zu behandelnden Probleminstanzen industrieller Anwendungsfälle noch prototypisch. Dies liegt daran, dass durch die aktuellen Hardwareeigenschaften Quantenschaltkreise (Softwarelösungen) noch sowohl in ihrer Breite als auch in ihrer durchführbaren Tiefe beschränkt sind (siehe Kapitel 1.1). Die Breite ist hierbei limitiert durch die zur Verfügung stehende Anzahl an Qubits (siehe Infobox 1), die Tiefe durch die Zuverlässigkeit der Gatteroperationen und der Kohärenzzeit (siehe Kapitel 1.2.4). Wie im einleitenden Kapitel 1.1 dargestellt, befinden sich diese Parameter über das gesamte Hardwarespektrum hinweg in stetiger und rapider Verbesserung. Wissenschaft, Wirtschaft und Politik investieren hier Hand in Hand signifikant in angewandte Forschung und Entwicklung.

Es ist jedoch zu erwarten, dass es selbst bei gleichbleibend hohem Entwicklungstempo noch einige Jahre dauern wird, bis die in Kapitel 0 vorgestellten Anwendungen praxisrelevante Problemgrößen effizient lösen können. Zudem ist für die hybriden Methoden der NISQ-Ära nicht rigoros beweisbar, dass diese einen Quantenvorteil zeigen werden. Hierbei handelt es sich meist um heuristische Ansätze einen Quantenvorteil zu realisieren. Die vorgestellten Ergebnisse stellen allerdings bereits heute ein Proof-of-Concept dar, dass wir erste Anwendungsfälle mittels Quantentechnologie als

Minimalbeispiele lösen können. Dies gilt es nun zu performanten End-to-End Lösungen weiterzuentwickeln, welche schlussendlich einen Quantenvorteil aufweisen.

Es ist daher bereits heute für potenzielle industrielle Anwender essenziell, jene Entwicklung engmaschig zu verfolgen, um direkt mit der Technologie mitzulernen und sich durch den Paradigmenwechsel der Quantensoftware-Entwicklung nicht abhängen zu lassen. Dies schließt die auf Hardwareseite stattfindende Entwicklung mit ein, um hier den Markt fundiert bewerten zu können. Ein Unternehmen sollte laufend beobachten, wie sich die Entwicklungen im Hardware-Software-Codesign auf seine relevanten Anwendungsfälle auswirken. Weiter ist es zentral auf Entwickelnden-Ebene ein gewisses Know-how, beispielsweise in Kollaborationen mit (angewandten) Forschungspartnern in gemeinsamen Projektvorhaben aufzubauen, um den angestrebten Quantenvorteil direkt nutzbar umzusetzen, sobald er sich bietet.

Autoren: Dr. Florian Knäble (Fraunhofer IAO), Dr. Christian Tutschku (Fraunhofer IAO)

Kontakt: [florian.knaeble@iao.fraunhofer.de](mailto:florian.knaeble@iao.fraunhofer.de), [christian.tutschku@iao.fraunhofer.de](mailto:christian.tutschku@iao.fraunhofer.de)



### 3 DER SEQUOIA ANWENDUNGSBAUKASTEN – QUANTENCOMPUTING IN DER PRAXIS

Zu den grundlegenden Erfolgsprinzipien des Software-Engineerings zählen Abstraktion, Wiederverwendbarkeit und leistungsfähige Werkzeuge. Sie ermöglichen Arbeitsteilung und Spezialisierung bei der Entwicklung von Anwendungen und leisten hierdurch einen erheblichen Beitrag zur Erzielung von Zeit- und Kostenvorteilen. Abbildung 48 zeigt typische Abstraktionsebenen bei der Entwicklung von Anwendungssoftware. Können Anwendungsentwickler\*innen auf einen leistungsfähigen Anwendungsbaukasten zurückgreifen, sinken Softwarefehler und die zur Realisierung einer Anwendung benötigte Zeit erheblich. Muss ein Unternehmen zur Realisierung einer Anwendung mehr Abstraktionsebenen adressieren, so steigt die benötigte Entwicklungszeit und in der Regel auch die Kosten.

#### Software-Engineering

Zielorientierte Bereitstellung und systematische Verwendung von Prinzipien, Methoden und Werkzeugen für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von umfangreichen Softwaresystemen.

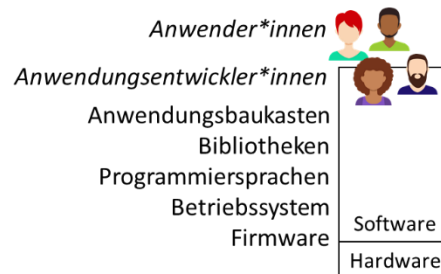


Abbildung 46: Abstraktionsebenen im Software-Engineering

Da Quantencomputer in den letzten Jahren einem breiteren Kreis von Anwender\*innen zugänglich wurden und sich mittelfristig eine kommerzielle Nutzung abzeichnet, hat sich die Weiterentwicklung insbesondere auf den unteren und mittleren Abstraktionsebenen erheblich beschleunigt. Anwendungsbaukästen wie Qiskit etc. unterstützen erste Problemstellungen auf hoher Abstraktionsebene, sind jedoch noch zu unflexibel, um eine größere Zahl realer Anwendungen darauf abbilden zu können. SEQUOIA stellt für die im Projekt adressierten/demonstrierten Use Cases die notwendige flexible Unterstützung durch den SEQUOIA Anwendungsbaukasten bereit.

### 3.1 Quantensoftware-Engineering

Während die Programmierung von Quantencomputern seit einigen Jahren erforscht wird, steht die Forschung im Hinblick auf Software-Engineering noch ganz am Anfang. Die Definition des Begriffs Quantensoftware-Engineering sowie zugehöriger Ausbildungsinhalte erfolgt seit dem Jahr 2020 durch eine ganze Reihe von Workshops wie dem »Workshop on Quantum Software Engineering (Q-SE)« [S 1] im Rahmen der IEEE International Conference on Software Engineering (ICSE) oder dem »Workshop on Quantum Software Engineering & Technology (Q-SET)« [S 2], [S 3] im Rahmen der IEEE International Conference on Quantum Computing and Engineering (QCE). Fokussiertere Workshops sind der »Workshop on Quantum Software Architecture (QSA)« [S 4] im Rahmen der IEEE International Conference on Software Architecture (ICSA), der »Workshop on Quantum Software Analysis, Evolution and Reengineering (Q-SANER)« im Rahmen der IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER) und die International Conference on Quantum Software (QSW) [S 5] im Rahmen des *IEEE World Congress on SERVICES*.

Die Erarbeitung der Definition des Begriffs Quantensoftware-Engineering diskutiert vor allem die Frage in welchen Aspekten Quantencomputing die einzelnen Teilgebiete des Software-Engineerings beeinflusst und neue Vorgehensweisen und Technologien erfordert. Nachfolgend werden diese Unterschiede und ihre Einflüsse auf das Software-Engineering ausführlicher erläutert.

Quantencomputing stellt keine Alternative zu klassischen Rechnerarchitekturen dar, sondern eine Ergänzung. Zum einen entfaltet Quantencomputing seine Überlegenheit nur bei der Lösung sehr spezifischer Probleme. Zum anderen sind in der aktuellen NISQ-Ära noch keine beliebig tiefen Quantenschaltkreise möglich (siehe Kapitel 1.2.4).

Diese beiden Umstände führten und führen zu einer Ausbildung sogenannter hybrider Ansätze, bei der klassisches Computing und Quantencomputing sich gegenseitig abwechseln (siehe Abschnitt 1.4.4 Variationelle Quantenalgorithmen). Entsprechend ist die zentrale Herausforderung für das Software-Engineering die möglichst nahtlose Integration beider Paradigmen. Im Gegensatz zu klassischer spezialisierter Hardware (GPU, FPGA, ASIC etc.) ist der konzeptionelle Bruch durch die dem Quantencomputing zugrundeliegenden Effekte der Quantenmechanik (siehe Kapitel 1.2) erheblich größer. So folgt die Codierung von Information in Qubits meist einer völlig anderen Logik als in klassischen Bits. Bei jedem Datentransfer zwischen klassischer Rechnerarchitektur und einem Quantencomputer ist somit eine komplexe Transformation notwendig. Diesen Overhead zu reduzieren, um die Effizienz von hybriden Ansätzen zu gewährleisten, ist eine beispielhafte Herausforderung.

Aus dem erheblichen konzeptionellen Unterschied gegenüber klassischen Computern und der gegenwärtigen Situation, in der Quantencomputing aus einer stark spezialisierten Forschungsdisziplin in die breite Anwendung(-sforschung) übergeht, ergeben sich einige Aufgabenfelder des Quantensoftware-Engineerings, in denen keine adäquaten Lösungen aus dem Software-Engineering bereitstehen.

### 3.1.1 Vorgehensmodelle

Für die Entwicklung einer Softwareanwendung hat das Software-Engineering verschiedene Vorgehensmodelle entwickelt. Diese variieren im Grad ihrer Formalität und können eher sequentiell oder eher evolutionär ausgestaltet sein. Ihnen gemein ist jedoch eine Unterteilung in Entwicklungsphasen, die (noch erweitert um eine Betriebs- oder Wartungsphase) zu einem allgemeinen Software-Lebenszyklus zusammengefasst werden können. Diesen skizzieren wir im Folgenden, bevor wir im nächsten Abschnitt darauf eingehen, wie sich ein entsprechender Quantensoftware-Lebenszyklus darstellen könnte.

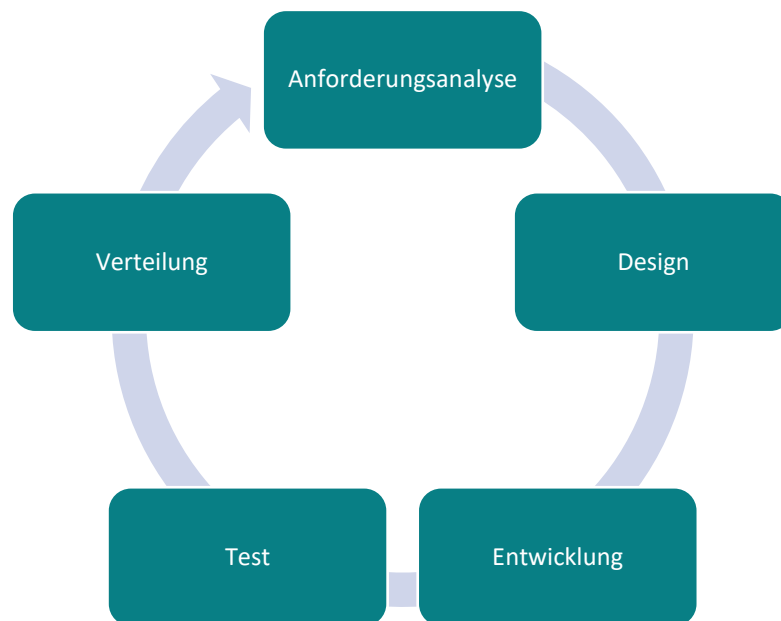


Abbildung 47: Phasen der Softwareentwicklung

Jeder Lebenszyklus beginnt mit einer *Anforderungsanalyse*, in der erhoben und dokumentiert wird, welche Aufgaben die zu erstellende Software zu erfüllen hat. Die hier detailliert niedergelegten Spezifikationen dienen in der *Designphase* dazu, die Softwarearchitektur festzulegen. Dies schließt die Aufteilung des Systems in Komponenten und deren Kommunikation untereinander ein, sowie die verwendeten Programmiersprachen und Frameworks. Außerdem müssen die Zielplattformen ausgewählt werden, auf denen die Software lauffähig sein soll. In der *Entwicklungsphase* wird dann gemäß dem Design die eigentliche Programmierung durchgeführt.

In der *Testphase* werden die Spezifikationen überprüft. Dies kann zum Beispiel automatische Tests kleiner Module, Integrationstests über mehrere Module und Lasttests auf Performanz umfassen.

In der *Verteilungsphase* wird die Software ausgerollt und zum Einsatz gebracht. Später auffallende Fehler werden in der *Wartungsphase* behoben. Diese Phase dauert den gesamten Betrieb über an und umfasst jegliche Veränderungen und Erweiterungen der Software, die durch Änderungen der Anforderungen im Zeitverlauf nötig werden können.

Bei der agilen Softwareentwicklung werden diese Phasen innerhalb fortlaufender Sprints immer wieder durchlaufen. Jeder anschließende Sprint kann auch als Wartungsarbeit an der bereits bestehenden Implementierung verstanden werden. Daher taucht die Wartungsphase im Kreislaufmodell in Abbildung 49 nicht explizit auf.

Im klassischen Wasserfallmodell findet jede Phase hingegen nur einmal statt.

### 3.1.2 Quantumsoftware-Lifecycle

Im Vergleich zum klassischen Vorgehensmodell sollen nun die für das Quantencomputing spezifischen Unterschiede herausgearbeitet werden. Dabei orientieren wir uns an den oben dargestellten Phasen.

Für einen Quantensoftware-Lebenszyklus, der über die hier dargestellte Anwenderebene hinaus auch auf technische Details eingeht, sei auf die umfassende Darstellung in [S 1] verwiesen. Die Auswirkungen auf das Zusammenspiel von Development und Operations (DevOps) werden zum Beispiel in [S 23] betrachtet.

#### 3.1.2.1 Anforderungsanalyse

Da die Anforderungsanalyse rein die Problemebene betrifft und noch unabhängig von der genauen Art der technischen Umsetzung ist, sind hier keine Einflüsse durch das Quantencomputing zu erwarten.

#### 3.1.2.2 Entwurf

Der Entwurf neuer Algorithmen wird in der Regel nicht als Kerndisziplin des Software-Engineerings betrachtet. Vielmehr adressiert Software-Engineering die Auswahl, Anpassung, Implementierung und Komposition von Algorithmen zu komplexen Softwarelösungen, um konkrete Anforderungen aus der Praxis zu erfüllen. Im Laufe der vergangenen Jahrzehnte sind eine Klassifikation von grundlegenden Problemen der Informatik und den zugehörigen Algorithmen zur Lösung dieser Probleme entstanden. Für viele Probleme gibt es dabei eine ganze Reihe von Algorithmen mit unterschiedlichen Eigenschaften, die meist gut verstanden sind. Zudem haben viele Algorithmen über die Zeit hinweg eine stetige Verbesserung erfahren. Im Gegensatz dazu decken die bisher

bekannten Quantenalgorithmen nur einen Teil der algorithmischen Problemstellungen ab, die bei der Entwicklung von Anwendungen zu lösen sind. Zudem sind die Eigenschaften von vielen Quantenalgorithmen im Hinblick auf Anwendungen in der Praxis bisher oft noch nicht gut verstanden. Quantensoftware-Engineering muss entsprechend den Entwurf neuer Algorithmen mitbetrachten, da auf absehbare Zeit bei der Entwicklung von Quantensoftware Problemstellungen auftreten werden, für welche passende Quantenalgorithmen fehlen.

Anwendungen für Quantencomputer bestehen zu einem großen Teil aus klassischen Softwarekomponenten, die mindestens Aufgaben wie Ein- und Ausgabe, Persistierung von Daten und die gesamte Ablauflogik der Anwendung bereitstellen. Diese klassischen Anteile werden um Quantensoftwarekomponenten ergänzt, die einzelne Quantenalgorithmen implementieren und Teile der Berechnungen auf Quantencomputern ausführen. Diese Berechnungen weisen jedoch einige Besonderheiten auf, wie z. B. die Notwendigkeit, für jede Ausführung neue Schaltkreise zu erzeugen, da Eingabedaten in Form von Initialisierungslogik für Qubits Teil des Schaltkreises sind. Das den meisten klassischen Rechnerarchitekturen zugrunde liegende Prinzip der Trennung von Daten und Programmcode greift somit nicht. Quantensoftware-Engineering muss dies beim Entwurf von Anwendungen durch entsprechende Modellierungs- und Spezifizierungstechniken sowie Entwurfsmuster unterstützen [S 6].

Beim Zusammenspiel von Quanten- und klassischen Softwarekomponenten zu einer Anwendung macht es der komplexe Entwurfsraum (Gesamtheit der Freiheitsgrade im Entwurf, die durch eine Entwurfsentscheidung ausgeprägt werden) sehr schwer, Annahmen zu qualitativen Eigenschaften, wie Performanz und Zuverlässigkeit, zu treffen. Insbesondere die Charakteristiken der NISQ-Ära (wenige sowie fehlerträchtige Qubits und begrenzte Zeit, in der die Kohärenz (siehe Kapitel 1.2.4) machen es notwendig, praxisrelevante Problemgrößen in kleinere Teile zu zerlegen.

Das Auffinden geeigneter Teilprobleme, bei deren Lösung die spezifischen Geschwindigkeitsvorteile des Quantencomputers gehoben werden können, ist die eigentliche Herausforderung des Entwurfs. Ebenso wie die Abbildung des Zusammenspiels zwischen klassischer und Quantensoftware in einem geeigneten hybriden Workflow. Zum Beispiel muss der Zustand des Quantencomputers für jede Teilberechnung vollständig neu initialisiert werden. Der Bedarf des Zerlegens in Teilberechnungen verbunden mit der fehlenden Möglichkeit Zustände über Berechnungen hinweg zu erhalten, hat beträchtliche Auswirkungen auf die Performanz einer Anwendung. Es kommt hinzu, dass aufgrund der stark unterschiedlichen Fehlercharakteristika von Quantencomputern und dem somit unterschiedlichen Bedarf zur Fehlerkompensation, Algorithmen bzw. deren Implementierungen sich sehr hardwareabhängig verhalten.

Methoden und Werkzeuge zur Analyse des komplexen Entwurfsraums von Anwendungen mit Quantensoftware-Komponenten im Hinblick auf die Vorhersage von Performanz und Zuverlässigkeit stellen daher eine wichtige Aufgabe des Quantensoftware-Engineering dar [S 7].

### 3.1.2.3 Programmierung

Bisher war die Programmierung von Quantencomputern eng verbunden mit deren Entwicklung. D. h. Personen, die Quantencomputer programmiert haben, waren in der Regel auch in deren Entwicklung eingebunden und hatten einen fachlichen Hintergrund in Quantenphysik oder ähnlichem. Aktuell findet der Übergang in die Anwendung statt, sodass zunehmend Personen ohne detaillierte Kenntnisse der physikalischen Grundlagen der Quantenmechanik sich der Programmierung von Quantencomputern zuwenden. Deshalb werden in der akademischen Ausbildung seit einigen Jahren Curricula erprobt, die ausloten, welche zusätzlichen Kompetenzen in die Informatikausbildung eingebracht werden müssen, um die Fähigkeit zur Programmierung von Quantencomputern zu vermitteln [S 8], [S 9], [S 10,] [S 12]. Inzwischen umfasst die Forschung ebenfalls erste Arbeiten, die die gegenwärtig verfügbaren Programmier-Frameworks für Quantencomputer aus didaktischer Sicht bewerten [S 13].

Die Programmierung einer Quantensoftware kann auf unterschiedlichen Abstraktionsebenen erfolgen (siehe Abbildung 50). Auf niedriger Ebene arbeitet man bei gatterbasierten Quantenprozessoren mit sogenannten »Quantenschaltkreisen«, die festlegen, in welcher Reihenfolge welche Quantengatter auf welche Qubits angewandt werden. Dagegen existieren auf höherer Ebene Anwendungsbaukästen, die für einzelne Teile von Algorithmen oder für ganze Algorithmen zugehörige Quantenschaltkreise generieren können.

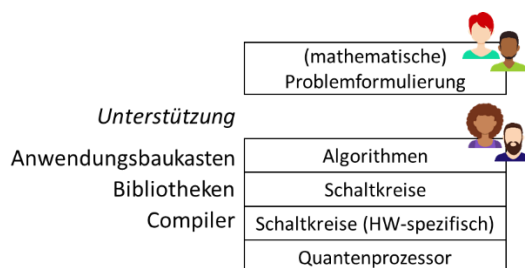


Abbildung 48: Abstraktionsebenen der Programmierung von Quantencomputern. Entsprechend der Abstraktionsebene ändern sich die Anforderungsprofile für die Programmierung. Für die Erstellung von Schaltkreisen ist ein detailliertes Verständnis der physikalischen Eigenschaften der einzelnen Quantengatter sowie der durch die Gatter realisierten Effekte auf Qubits notwendig. Im Gegensatz dazu kann auf Ebene von Anwendungsbaukästen mit wesentlich weniger Detailkenntnis der physikalischen Grundlagen programmiert werden.

Aktuell sind die durch Anwendungsbaukästen bereitgestellten Implementierungen von Algorithmen oftmals für eine konkrete Problemstellung optimiert. Sie weisen entsprechend wenig Flexibilität auf und erzeugen bereits bei einer leicht variierten Problemstellung sehr ineffiziente Quantenschaltkreise. Generische Implementierungen von Quantenalgorithmen, die für unterschiedliche Ausprägungen einer Problemstellung in der Lage sind, effiziente Schaltkreise zu erstellen, werden erst langsam Verbreitung finden. Das SEQUOIA-Konsortium wird für ausgewählte Anwendungsfälle derartige flexible Implementierungen demonstrieren.

Auf der Ebene der Quantenschaltkreise stellen die meisten Programmier-Frameworks Bibliotheken mit Schaltkreisbausteinen bereit, die bestimmte Funktionen realisieren sowie zu Algorithmen und Anwendungen zusammengesetzt werden können. Die Quantenschaltkreise werden in den meisten Frameworks in einem universellen Format definiert und dann für die Ausführung auf einem konkreten Quantenprozessor in prozessorspezifische Schaltkreise übersetzt. Dabei werden sowohl die für den Prozessor verfügbaren Gatter als auch dessen Topologie berücksichtigt. Compiler führen diesen Schritt automatisch aus. Allerdings kann durch die dynamische Entwicklung der Quantenprozessoren die Effizienz der Übersetzung stark variieren.

#### 3.1.2.4 Test und Validierung

Das Testen und Validieren von Software für Quantencomputer birgt mehrere große Herausforderungen. So gibt es nur sehr beschränkte Möglichkeiten, Zwischenzustände eines Quantenschaltkreises auszuwerten, da Qubits beim Auslesen ihre quantenmechanischen Eigenschaften, (siehe Abschnitt 1.2.1) wie Superposition und Verschränkung, verlieren [S 14]. Deshalb ist ein Quantencomputer als »Blackbox« aufzufassen, der eine Anfrage annimmt, nicht einsehbare Berechnungen auf Basis der quantenmechanischen Eigenschaften durchführt und am Ende eine Antwort liefert. Dieses grundlegende Prinzip des Quantencomputings hat zur Konsequenz, dass für Test und Validierung nur Techniken eingesetzt werden können, die eine »Blackbox-Sicht« auf die Quantensoftwarekomponenten einnehmen. Hierbei kann das Verhalten einer Quantensoftwarekomponente lediglich auf Basis ihrer Ein- und Ausgabedaten bewertet werden und nicht durch eine Analyse der genauen Abläufe innerhalb der Komponente.

Darüber hinaus liefert das Auslesen von Qubits bedingt durch die Natur des quantenmechanischen Messprozesses ein inhärent zufälliges Ergebnis. Durch wiederholte Ausführung einer Berechnung entsteht eine Wahrscheinlichkeitsverteilung der Ausgabezustände, aus der dann das Ergebnis der Berechnung abgeleitet werden muss. In der aktuellen NISQ-Ära weisen Quantencomputer zudem erhebliche Fehlercharakteristiken auf. Beispielsweise führen einzelne Quantengatter in einer

Quantenschaltung nicht exakt die vorgesehene Zustandsänderung von Qubits durch oder Qubits weichen bereits nach sehr kurzen Zeitspannen von ihrem ursprünglich gesetzten Zustand ab.

Im Kontext der genannten Eigenschaften von Quantencomputern wird klar, dass eine Implementierung eines Quantenalgorithmus schwer als korrekt bewertet werden kann. Nur wenn diese Implementierung tolerant gegenüber den Fehlercharakteristiken eines bestimmten Quantencomputers ist, liefert der Quantenalgorithmus Ergebnisse, die es ermöglichen, die falschen von den richtigen Ergebnissen zu separieren.

### 3.1.2.5 (Software-)Verteilung

Auch wenn einige Hardwareanbieter bereits in Aussicht stellen, Quantencomputer ausliefern zu wollen, auf die der Anwender direkt selbst zugreifen kann, erfolgt der Zugriff aktuell immer noch über Cloud Systeme, an die man die entwickelten Schaltkreise als einzelne Jobs absendet. Der klassische Teil der Software, der wie gewohnt verteilt werden kann, muss also nur mit den dafür notwendigen Informationen über diese Endpunkte ausgestattet werden. Die (temporäre) Installation der Quantenschaltkreise für die Ausführung eines Jobs auf dem Backend wird hingegen komplett vom Framework des jeweiligen Herstellers übernommen.

### 3.1.2.6 Wartung

Die Wartung von Quantensoftware unterscheidet sich von Software für konventionelle Hardwarearchitekturen vor allem durch das sich rasch ändernde Hardwareumfeld und die Tatsache, dass Quantencomputer nur über Clouddienste zur Verfügung gestellt werden.

Es werden verschiedene Realisierungen von Quantencomputern parallel erforscht, die auf unterschiedlichen Technologien (Supraleiter, Ionenfallen, Photonen etc., siehe Kapitel 1.3) basieren. Nicht alle Software Development Kits harmonieren mit jedem Ansatz und Stand heute ist unklar, ob sich alle Ansätze durchsetzen werden. Auch innerhalb einer Technologie gibt es eine vergleichsweise rasche Abfolge von Prozessorgenerationen. Um von verbesserter Performanz oder verbesserter Fehlerkorrektur zu profitieren, kann unter Umständen ein Update der Software von Nöten sein.

Auch die aktuell zur Verfügung stehenden Software Development Kits für die Quantenprogrammierung ändern sich tendenziell schneller und volatiler als ihre konventionellen Gegenstücke. Damit kann die Adaption einer neuen Version auch Nacharbeiten an der Codebasis nach sich ziehen.

Schlussendlich sind die aktuell prominenten variationellen Algorithmen (siehe Abschnitt 1.4.4) noch in der Erforschungsphase und damit per se volatil. Deren Implementierungen



werden deshalb ebenfalls einem ständigen Verbesserungsprozess unterworfen sein. All diese Faktoren lassen tendenziell kürzere Wartungsintervalle erwarten und unterstreichen damit nochmals die gesteigerte Bedeutung einer hohen Wartbarkeit.

### 3.1.3 Bedeutung von Simulatoren

Es besteht grundsätzlich die Möglichkeit einen Quantenalgorithmus auf einem konventionellen Computer zu simulieren. Die Anwendung eines Quantenschaltkreises auf einen Anfangszustand entspricht mathematisch einer Matrixmultiplikation. Die Größe der entsprechenden Matrix wächst allerdings exponentiell mit der Anzahl der simulierten Qubits und demgemäß auch der Bedarf an Arbeitsspeicher. Während man bei 10 Qubits noch im Bereich von Megabyte ist, um die entsprechende Matrix darzustellen, erreicht man bei 20 Qubits schon den Bereich von Terabyte. Tabelle 3 in Kapitel 1 zeigt den genauen Zusammenhang zwischen der Anzahl von Qubits und dem Speicherbedarf für einen Quantenzustand. Man beachte, dass man für den Bedarf der Matrix, den Bedarf für den Zustand quadrieren muss.

Es gibt auch effizientere Simulationsmethoden [S 15], welche nicht die gesamte Matrix gleichzeitig im Speicher benötigen, aber das Wachstum bleibt exponentiell.

Für Algorithmen mit einer niedrigen zweistelligen Zahl von Qubits ist eine Simulation aktuell noch gut darstellbar und auch deutlich günstiger als einen echten Quantenrechner zu verwenden. Eine Simulation ist zudem frei von der bereits erwähnten Fehleranfälligkeit aktueller Quantencomputer. Auch die probabilistische Problematik wird hinfällig, wenn man einen Simulator verwendet. Dieser kann sämtliche Amplituden des Ergebnisvektors ausgeben, die man physikalisch durch wiederholte Messungen herausarbeiten müsste. Damit lässt sich leichter erkennen, ob der Algorithmus tatsächlich so abläuft wie konzipiert. Wiederum im Vergleich mit der Rechnung auf einem echten Quantencomputer kann dies außerdem das Verständnis dafür schärfen, wie sich Fehler auswirken und Möglichkeiten eröffnen, daraus Mitigationsmethoden abzuleiten.

Bei der Entwicklung von Quantenalgorithmern ist es also angebracht, die Ergebnisse immer zuerst mit einem Simulator zu validieren, bevor man zur Rechnung auf einem echten Quantenrechner übergeht. Als Zwischenschritt gibt es auch noch die Möglichkeit, das Fehlerverhalten eines bestimmten Quantenrechners in die Simulation miteinzubeziehen. Wie Algorithmen auf Problemgrößen verifiziert werden, die einer Simulation nicht mehr zugänglich sind, ist allerdings eine offene Forschungsfrage.

### 3.1.4 Cloud-Backends

Derzeitig stellen alle Anbieter von Quantencomputern ihre Hardware als Cloud-Lösung bereit. Um eine Rechnung auf einem Quantencomputer auszuführen ist es, ähnlich wie

beim Hochleistungsrechnen, notwendig, das erstellte Programm in eine Warteschlange einzureihen, die dann vom Cloud-Backend abgearbeitet wird. Da man keinen direkten Einfluss auf die Zeit hat, die bis zur Ausführung des Programms verstreicht, werden bestehende Workflows so unterbrochen.

Insbesondere bei hybriden Programmen, die ein Wechselspiel von klassischem und Quantum Computing benötigen, kann die Ausführungsdauer so beträchtlich steigen. Eine Messung der Performanz wird deutlich erschwert. Eine Lösung hierfür besteht darin, das gesamte Programm (klassischer Teil wie Quantenteil) in einer Laufzeitumgebung in der Cloud ausführen zu lassen. Dies garantiert immer noch nicht eine durchgängige Ausführung, aber eine drastische Reduktion der Wartezeiten. Der im SEQUOIA-Baukasten enthaltene Ausführungsservice, der in Abschnitt Ausführungs-Modul beschrieben wird, unterstützt ebenfalls beim Umgang mit Cloud-Backends.

Autoren: Oliver Denninger (FZI Forschungszentrum Informatik), Dr. Florian Knäble (Fraunhofer IAO), Daniel Mayer (Fraunhofer IAO),

Kontakt: [denninger@fzi.de](mailto:denninger@fzi.de), [florian.knaeble@iao.fraunhofer.de](mailto:florian.knaeble@iao.fraunhofer.de),  
[daniel.mayer@iao.fraunhofer.de](mailto:daniel.mayer@iao.fraunhofer.de)

## 3.2 Der SEQUOIA Werkzeugkasten - High-Level Entwickler API

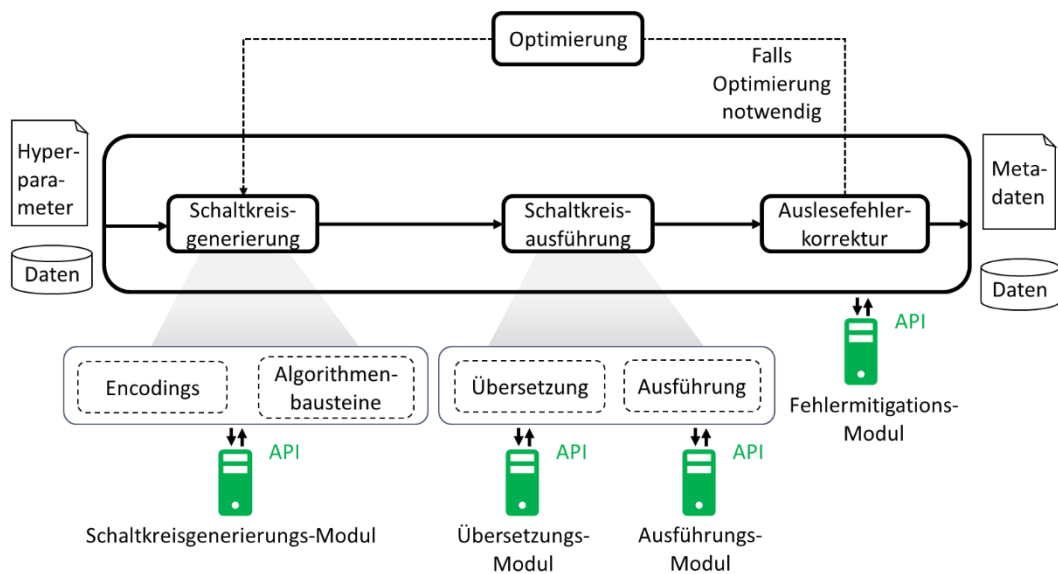


Abbildung 49: Schematische Darstellung der Verwendung des Werkzeugkastens

Wie weiter oben bereits angedeutet, steht und fällt die Produktivität der Softwareentwicklung mit der Verfügbarkeit von Software-Baukästen, die wiederverwendbare Implementierungen zur Lösung gängiger Problemklassen enthalten. Im Rahmen des SEQUOIA-Projektes werden daher vorhandene Software-Bausteine zusammengetragen und bewertet. Ergänzend wurden eigene Komponenten entwickelt, welche zusammen mit den vorhandenen in einen Werkzeugkasten integriert wurden. Dabei wurde besonderes Augenmerk daraufgelegt, dass die Komponenten anbieterunabhängig sind, sodass Interoperabilität mit allen gängigen Frameworks gewährleistet ist.

Der Werkzeugkasten gliedert sich in die vier verwendeten Application Programming Interfaces (APIs) zur Umsetzung von Quantenalgorithm, welche in Abbildung 51 dargestellt sind. Im Folgenden werden diese kurz dargestellt.

### 3.2.1 Schaltkreisgenerierungs-Modul

Bei der Bearbeitung der in Kapitel 0 beschriebenen Use Cases sind Schaltkreisimplementierungen von Quantenalgorithm entstanden, welche als Teil des Werkzeugkastens veröffentlicht werden. Dazu gehören zum Beispiel die in Abschnitt 3.3 vorgestellte Lösung eines modifizierten Traveling-Salesperson-Problems und die Implementierung eines Variational Quantum Linear Solvers zur Lösung linearer Gleichungssysteme. Diese Schaltkreisimplementierungen werden über eine API zur weiteren Verwendung zur Verfügung gestellt, sodass Schaltkreise für betreffende Probleme automatisch generiert und als Bausteine in anderen Algorithmen verwendet werden können. Weitere enthaltene Algorithmen(-bausteine) sind die Quantum Fourier

Transformation (*QFT*), Quantum Phase Estimation (*QPE*), Quantum Approximate Optimization Algorithm (*QAOA*) und Quantum Algorithm for Linear Systems of Equations (*HHL*).

Um Daten auf dem Quantenrechner zu verarbeiten, müssen diese enkodiert werden. Dies ist vergleichbar mit der Binärcodierung einer Zahl für die Verarbeitung auf einem klassischen Computer. Zur Schaltkreisgenerierung stehen daher auch Bausteine bereit, um klassische Daten auf alle gängigen Arten (z. B. Basis-Enkodierung, Winkel-Enkodierung, Amplituden-Enkodierung) als Quanten-Daten zu enkodieren. Zum Beispiel benötigt der HHL-Algorithmus seinen Input in Basis-Enkodierung, stellt seinen Output hingegen in Amplituden-Enkodierung zur Verfügung. Eine Übersicht über Enkodierungen für Quantenalgorithmen findet sich zum Beispiel in [S 17].

### 3.2.2 Übersetzungs-Modul

Aktuell existieren diverse unterschiedliche Programmiersprachen und Frameworks zur Implementierung von Quantenalgorithmen, welche eine starke Abhängigkeit zum jeweiligen Hersteller zeigen. Im Allgemeinen existieren nicht für jede Programmiersprache auch die notwendigen Compiler, um die Schaltkreise für die Hardware aller Hersteller in ausführbaren Code zu kompilieren. Um Implementierungen in unterschiedlichen Frameworks verwenden zu können, existieren eine Reihe von Übersetzern, die Schaltkreise zwischen unterschiedlichen Sprachen konvertieren können. Diese werden im Übersetzungsmodul integriert und über eine Schnittstelle bereitgestellt. Somit können unter anderem die durch das Schaltkreisgenerierungs-Modul erzeugten Schaltkreise in mehreren Sprachen verfügbar gemacht werden. Dies erleichtert die Ausführung der Schaltkreise auf der Hardware unterschiedlicher Hersteller.

### 3.2.3 Ausführungs-Modul

Die Ausführungskomponente erlaubt es, Quantenschaltkreise über eine Schnittstelle auf unterschiedlichen Quantencomputern auszuführen. Diese können von unterschiedlichen Herstellern stammen und die Schaltkreise in unterschiedlichen Sprachen vorliegen. Daher kann es notwendig sein, dass Schaltkreise vor der Ausführung mit Hilfe des Übersetzungs-Moduls in eine kompatible Sprache übersetzt werden. Damit wird gewährleistet, dass die vorliegenden Quantenschaltkreise auf einer geeigneten QPU ausgeführt werden können. Die als Service gestaltete Ausführungskomponente übermittelt die Schaltkreise zur Ausführung an den jeweiligen Provider und ruft die Ergebnisse ab, sobald sie bereitstehen. Hierzu werden die auszuführenden Schaltkreise in vom Provider unterstützte Ausführungs-Jobs verpackt. Somit wird eine einfache und kompakte Ausführung mit minimaler Wartezeit sichergestellt.

### 3.2.4 Fehlermitigations-Modul

Das Fehlermitigations-Modul hilft dabei, den Einfluss der in der NISQ-Ära typischen Fehler bei der Ausführung von Quantenschaltkreisen zu reduzieren. Eine ausführlichere Diskussion zu Fehlern in der NISQ-Ära und den Fehlerarten findet sich in Abschnitt 1.5. Der Fokus des Fehlermitigations-Moduls liegt gegenwärtig auf der Abmilderung (Mitigation) von Auslesefehlern. Über die Schnittstelle können hierzu Kalibrierungsdaten für die Quantencomputer unterschiedlicher Hersteller generiert werden. Dazu wird eine Reihe von speziell präparierten Kalibrierungsschaltkreisen zur Messung von Fehlerraten auf der Hardware ausgeführt. Anhand der gemessenen Fehlerraten wird anschließend ein Mitigationsoperator erstellt, der auf die Ausführungsergebnisse angewendet werden kann, um aufgetretene Auslesefehler abzumildern.

Bei den in der NISQ-Ära typischen variationellen Quantenalgorithmen werden parametrisierte Schaltkreise in einer Schleife immer wieder ausgeführt und lediglich die Parameter verändert. Damit die Kalibrierung nicht für jede Iteration wiederholt werden muss, speichert das Fehlermitigations-Modul die Kalibrierungsdaten und Mitigationsoperatoren in einer Datenbank. Somit können diese über einen längeren Zeitraum, in dem sich die Fehlerraten der Hardware nicht wesentlich verändern, wiederverwendet und auch anderen Benutzern zur Verfügung gestellt werden.

### 3.2.5 Automatisierung von Quantenanwendungen

Die beschriebenen Module können, wie in Abbildung 51 dargestellt, zur Umsetzung von Quantenalgorithmen verwendet werden. Wie bei klassischen Anwendungen bereits üblich, können auch (hybride) Quantenanwendungen von Automatisierung profitieren. Eine etablierte Technologie zur Automatisierung von komplexen Anwendungen sind sogenannte Workflows. In einem Workflowmodell werden die benötigten Arbeitsschritte, deren Anordnung und der Datenfluss beschrieben. Anschließend kann dieses Modell mittels einer Workflow-Engine automatisiert ausgeführt werden. Hierdurch ergeben sich Vorteile wie eine hohe Skalierbarkeit und Robustheit der modellierten Anwendungen. Workflows können an beliebiger Stelle unterbrochen und fortgesetzt, sowie transaktional abgesichert werden. Aufgrund des kontinuierlich überwachten Datenflusses sind alle Abläufe einsehbar und nachvollziehbar.

In Workflows lassen sich unterschiedlichste Anwendungstypen flexibel miteinander verknüpfen. Insbesondere wurde die Workflowmodellierungssprache BPMN durch QuantME [S 20] erweitert, sodass auch für Quantenalgorithmen typische Abläufe, wie die Ausführung von Schaltkreisen, einfach modelliert werden können. Hierzu eignen sich die zuvor vorgestellten Module des Werkzeugkastens, welche über die APIs in Workflows eingebunden werden können.

### 3.2.6 Erweiterbarkeit

Alle Komponenten des Baukastens sind Open Source und werden auf GitHub zur Verfügung gestellt. Bei der Konzeptionierung der Module wurde auf eine einfache Erweiterbarkeit geachtet, sodass diese für unterschiedlichste Anwendungsfälle angepasst werden können. Auch eine Übernahme von Änderungen in den Werkzeugkasten (Pull Request) wird möglich sein.

### 3.2.7 Relation zum NISQ Analyzer

Der NISQ Analyzer [S 16] ist ein im Rahmen des assoziierten PlanQK<sup>11</sup>-Projektes entwickeltes Werkzeug, das die Auswahl von QPUs zur Ausführung von Quantenschaltkreisen unterstützt und erleichtert. Dieser weist Synergien zum SEQUOIA-Werkzeugkasten auf, weshalb darauf geachtet wurde, dessen Module kompatibel zu gestalten.

Mit der hohen Anzahl vorhandener Quantencomputer ist es schwierig festzustellen welches die optimale QPU zur Ausführung eines gegebenen Quantenalgorithmus darstellt. Mit dem Quantenvolumen [S 21] wurde eine Orientierungsgröße für die Performanz von QPUs geschaffen, die über die reine Anzahl von verfügbaren Qubits hinausgeht. Jedoch ist die QPU mit dem höchsten Quantenvolumen nicht notwendigerweise optimal für einen gegebenen Schaltkreis.

Um die beste QPU für die Ausführung eines konkreten Schaltkreises auszuwählen, kann der NISQ Analyzer genutzt werden. Dieser verfügt über eine Auswahl von QPUs für die er die Tiefe und Breite des kompilierten Schaltkreises automatisch ermittelt und daraufhin eine QPU empfehlen kann. Der NISQ Analyzer stößt dann die Ausführung auf dem Cloud-Backend des jeweiligen Herstellers an und liefert das entsprechende Resultat zurück.

Der NISQ Analyzer greift also ebenfalls auf die Übersetzung und Ausführung von Schaltkreisen für unterschiedliche Hardware zurück. Daher werden das Übersetzungs- und Ausführungs-Modul so umgesetzt und erweitert, dass diese auch an den NISQ Analyzer angebunden werden können.

Autoren: Martin Beisel (Universität Stuttgart IAAS),

Felix Truger (Universität Stuttgart IAAS)

Kontakt: [martin.beisel@iaas.uni-stuttgart.de](mailto:martin.beisel@iaas.uni-stuttgart.de), [felix.truger@iaas.uni-stuttgart.de](mailto:felix.truger@iaas.uni-stuttgart.de)

---

<sup>11</sup> PlanQK (Plattform und Ökosystem für Quantenunterstützte Künstliche Intelligenz) ist ein vom Bundesministerium für Wirtschaft und Klimaschutz gefördertes Partnerprojekt von SEQUOIA

### 3.3 Beispiel anhand eines Use Cases

Der Einsatz von Quantenrechnern der NISQ-Ära für praktisch relevante Probleme ist bisher aufgrund deren stark beschränkter Größe und Fehleranfälligkeit nur in sehr speziellen Anwendungsgebieten (wie etwa der Quantenchemie) sinnvoll. Klassische Optimierungsprobleme, wie beispielsweise solche aus der Kombinatorik, bei denen aus einer exponentiell wachsenden Menge an Lösungsmöglichkeiten die beste gewählt werden soll, können schon für kleine Instanzen nicht in ihrer Gänze auf heutige Quantenrechner abgebildet werden. Insbesondere derartige Optimierungsprobleme sind jedoch auf klassischen Rechnern ausschließlich unter hohem zeitlichem Aufwand und mit aufwendiger Hardware lösbar, da für eine exakte Lösung bisher kein effizienter Algorithmus bekannt ist. Zwar existieren für klassische Rechner Algorithmen, die gute approximative Lösungen liefern und die auch effizient berechnet werden können - dennoch versprechen Quantenalgorithmen in dieser Domäne einen Laufzeitvorteil erzielen zu können.

Ein Beispiel für ein kombinatorisches Optimierungsproblem ist das Travelling Salesperson Problem, kurz TSP. Dabei möchte ein Handlungsreisender, ausgehend von einem bestimmten Startpunkt, diejenige Rundreise durch eine gegebene Anzahl an Städten ermitteln, bei der die dabei zurückgelegte Wegstrecke minimal wird. Die Berechnungsgrundlage dafür ist eine Tabelle, in der die Distanzen zwischen den Städten vermerkt sind. Während das TSP bereits in dieser einfachen Form praktische Anwendung in der Tourenplanung und in der Logistik findet, so sind insbesondere spezialisierte Varianten, die auch zusätzliche Einschränkungen in der Problemstellung beinhalten können, weit verbreitet. Unter diese zusätzlichen Einschränkungen fallen beispielsweise ein maximaler Rohstoffverbrauch (wie Treibstoff), der auf den Wegstrecken anfällt oder Zeitfenster, außerhalb derer bestimmte Städte nicht angefahren werden können. An der Schwierigkeit des Problems ändern derartige Einschränkungen im Großen und Ganzen allerdings nichts.

Auf einem klassischen Rechner kann ein TSP mittels dynamischer Programmierung gelöst werden. Die Idee dabei ist, dass man ein TSP mit einer bestimmten Anzahl Städte durch Weglassen einer Stadt auf eine kleinere Probleminstanz reduziert, zunächst diese löst und dann zu dieser Lösung eine weitere Stadt zur Tour hinzuzufügt. Diese Rekursion wird so lange aufgelöst, bis der Trivialfall für zwei Städte eintritt, der einfach gelöst werden kann. Damit im Verlauf der Berechnung keine Zwischenergebnisse mehrfach bestimmt werden, wird eine Tabelle angelegt, in der diese Zwischenergebnisse festgehalten werden. Diese Tabelle wird dann Schritt für Schritt befüllt - in der letzten Zelle kann die Lösung für das vollständige Problem ausgelesen werden. Schematisch ist dieses Vorgehen im linken Teil von Abbildung 52 dargestellt. Dadurch, dass bei der Berechnung keine Redundanzen entstehen, wird die Laufzeit im Vergleich zu simpler

Rekursion stark reduziert. Da man jedoch durch die Tabelle zusätzlichen Speicheraufwand erhält, handelt es sich um eine Abwägung zwischen Laufzeit und Speicherverbrauch.

Kleine Instanzen des TSP können exakt oder approximativ auf dem IBM Quantum System One in Ehningen gelöst werden. Insbesondere Approximationen mittels des Quantum Approximate Optimization Algorithm (QAOA) sind von einer ausgiebigen Anzahl an wissenschaftlichen Arbeiten gestützt. Er wurde eingeführt in [S 18] und erweitert in [S 19]. Im ersten Kapitel dieser Studie findet sich unter Absatz 1.4.4.1 ebenfalls eine Darstellung. QAOA ist ein sogenannter hybrider Quantenalgorithmus, der aus zwei Teilen besteht: Ein Teil wird auf einem klassischen Rechner ausgeführt, der andere auf einem Quantenrechner.

Der Teil, der auf dem Quantenrechner ausgeführt wird, besteht aus zwei parametrisierten, sich abwechselnden Operatoren, die entweder die aktuelle Lösung bewerten (beim TSP also beispielsweise die Länge der Tour) oder diese in eine neue Lösung überführen. Werden diese Operatoren sehr häufig mit optimalen Parametern ausgeführt, liefert diese Berechnung theoretisch eine optimale Lösung für das eingegebene Problem. In der Praxis kann man diese Operatoren jedoch nicht beliebig oft ausführen und auch die optimalen Parameter dafür werden vom klassischen Löser nur angenähert. Deshalb wird die optimale Lösung nur mit einer gewissen Wahrscheinlichkeit ermittelt. Für andere, unter Umständen schlechtere Lösungen, besteht demnach ebenfalls die Möglichkeit als Ergebnis zurückgegeben zu werden.

Der Teil von QAOA, der klassisch berechnet wird, ist ein herkömmliches Optimierungsproblem, das die Parameter für die Quantenoperatoren ermittelt. Beide Teile wechseln sich also unter sich ständig verändernden, vom klassischen Algorithmus ausgewählten, Parametern ab. Infolgedessen entscheidet der klassische Rechner, wann ein Optimum erreicht wurde, und der Algorithmus terminiert. Für große, praktisch relevante TSP-Instanzen haben aktuelle Quantenrechner allerdings nicht die notwendige Anzahl von Qubits, um die dazugehörigen Lösungen ausdrücken zu können.

In Abbildung 52 links ist schematisch die herkömmliche dynamische Programmierung dargestellt. Von einem einfach zu berechnenden Startpunkt aus wird eine Tabelle in kleinen Schritten befüllt. Weil die jeweiligen Schritte eine minimale Größe haben (hier 1), kann der nächste Tabelleneintrag direkt bestimmt werden.

Rechts steht dazu im Vergleich die dynamische Programmierung, wenn sie zur Aufspaltung eines Problems in kleinere Teilprobleme der Größe  $k$  verwendet wird. Die Schritte zwischen den Tabelleneinträgen sind so groß, dass sie für sich genommen eine neue Probleminstanz darstellen.



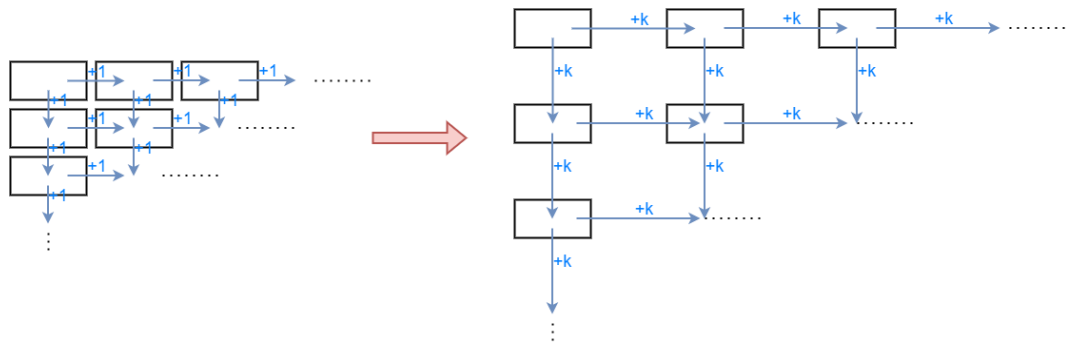


Abbildung 50: Dynamische Programmierung

Ein möglicher Ansatz ist wiederum dynamisches Programmieren zu verwenden, um große Probleme in kleinere Teilprobleme aufzuspalten und sie so für den Quantenrechner berechenbar zu machen. Dabei kann man sich einer Modifikation des Algorithmus bedienen bei der, im konkreten Fall für TSP, nicht eine einzige Stadt pro Tabelleneintrag hinzugefügt wird, sondern mehrere. Diese Menge an Städten und deren dazugehörigen Distanzen untereinander bilden ein Teilproblem, das beliebig klein gewählt werden kann, um für den Quantenrechner beherrschbar zu werden. In der Quintessenz hat man also einen Algorithmus, der aus den Lösungen vieler kleiner, gleich großer Teilprobleme des TSPs die Lösung für ein großes Problem zusammensetzt, an der Hand. Sobald in der Zukunft für die Teilprobleme ein Quantenvorteil gegenüber der klassischen Lösung erreicht werden kann, könnte daraus auch ein Quantenvorteil für die Lösung des gesamten TSP entstehen.

Autoren: Simon Garhofer (Eberhard Karls Universität Tübingen),  
Oliver Denninger (FZI Forschungszentrum Informatik)

Kontakt: [simon.garhofer@uni-tuebingen.de](mailto:simon.garhofer@uni-tuebingen.de), [denninger@FZI.de](mailto:denninger@FZI.de)

## 4 WISSENSTRANSFER UND ÖKOSYSTEM

Zentraler Anspruch dieser Studie war die Ergebnispräsentation der im SEQUOIA Projekt mittels Quantentechnologie umgesetzten Anwendungsfälle (siehe Kapitel 2) sowie der dabei entwickelten Werkzeuge, Schnittstellen und Softwarekomponenten (siehe Kapitel 3). Wie in Kapitel 1.1 sowie Kapitel 2.1 angedeutet, wurden jene Ergebnisse im Schulterschluss mit dem etablierten SEQUOIA Unternehmensnetzwerk erarbeitet. Der Aufbau solcher Netzwerke mit Firmen aus unterschiedlichen Branchen sowie der kontinuierliche Wissenstransfer in jene Ökosysteme ist maßgeblich für die Verwertung, Verstetigung und damit den Erfolg eines jeden angewandten Forschungsprojektes. Nur ein gemeinsam wachsendes Ökosystem aus Wirtschaft, Wissenschaft und Politik kann langfristig die Etablierung, den zielorientierten Einsatz sowie die Souveränität einer jeden Zukunftstechnologie sicherstellen. Wissenstransfer und Vernetzung im Ökosystem stellen also zentrale KPIs für derartige Vorhaben dar. Wie jene Ziele erreicht bzw. wie Unternehmen Teil jener Aktivitäten werden können, wird im Folgendem exemplarisch am Beispiel des SEQUOIA Unternehmensnetzwerks dargestellt.

Im Fokus dieses Netzwerks steht der Austausch über aktuelle Entwicklungen des Quantencomputings (siehe Kapitel 1.1 sowie Kapitel 2.1) sowie die Erarbeitung von konkreten Nutzungsszenarien für Quantencomputer in der Wirtschaft (siehe Kapitel 2). Die Unternehmen bekommen dabei die Möglichkeit, sich untereinander zu vernetzen und gemeinsam mit dem Projektkonsortium ihre spezifischen Nutzungsszenarien für einen potenziellen Einsatz des Quantencomputers im eigenen Unternehmen zu erarbeiten. Interessierte Unternehmen werden somit auf den Einsatz von Quantencomputern vorbereitet (englisch Quantum Ready), indem frühzeitig das Bewusstsein für die Technologie des Quantencomputings (englisch Quantum Awareness) geschaffen wird: »From Quantum Awareness to Quantum Readiness«.

So wurden die gemeinsam erarbeiteten Projektergebnisse in insgesamt 95 Transferveranstaltungen sowohl dem Netzwerk, als auch der Fachöffentlichkeit kommuniziert bzw. verständlich gemacht. Dabei wurden im Rahmen von SEQUOIA Transferveranstaltungen ca. 6400 Teilnehmende erreicht bzw. in verschiedenen fachlichen Tiefen geschult. Von der Management Sicht bis zum Entwicklerwissen wurde die erarbeitete Expertise in zahlreichen Veranstaltungen mit unterschiedlichem Tiefgang vermittelt. Zusammenfassend ist dies in Abbildung 53 visualisiert. Neben den projektinternen Veranstaltungen wurden die erarbeiteten Projektergebnisse in etablierte Fraunhofer Weiterbildungsangebote integriert sowie in Partner-spezifischen Industrieveranstaltungen direkt transferiert. Dabei war das Projekt nicht nur auf nationaler, sondern insbesondere auch auf der internationalen Bühne vertreten. Die konkrete Liste der zugrundeliegenden Veranstaltungen ist der Projektwebsite zu entnehmen.

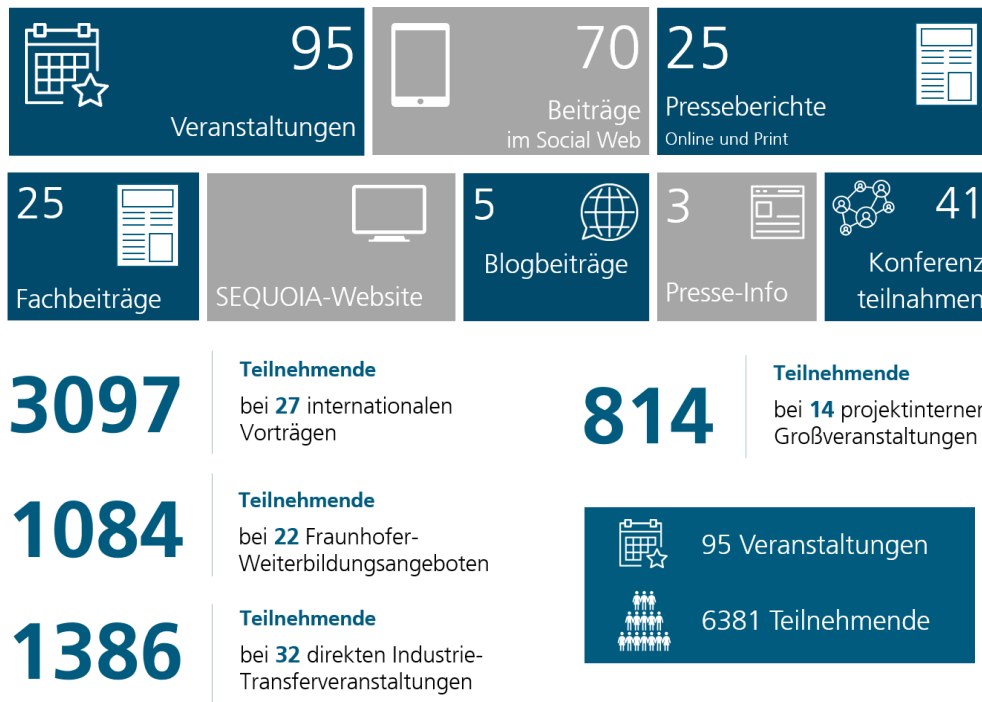


Abbildung 51: Wissenstransfer- und Öffentlichkeitsarbeit im SEQUOIA Projekt. Die konkrete Liste der zugrundeliegenden Veranstaltungen ist unter [WT1] zu finden

Um das nötige Know-How zur Nutzung von Quantencomputern wie dem IBM-Quantencomputer in Ehningen zielgruppenspezifisch zu vermitteln, entwickelte das Fraunhofer IAO und das Fraunhofer IAF im Rahmen des Kompetenzzentrums »Quantencomputing Baden-Württemberg« ein Schulungsangebot Quantencomputing (fraunhofer.de), welches dies über das Jahr 2022 erstmals systematisch getan hat. Als Leuchtturmprojekt für einen effizienten und zielorientierten Wissenstransfer soll dies im Folgenden repräsentativ vorgestellt werden.

Vermittelt wurden in jenem Programm fundierte Kenntnisse in Quantentechnologien und deren Programmierung sowie erste Erfahrungen mit realen Quantencomputern wie dem IBM Quantum System One in Ehningen (Grundlagenkurs Quantencomputing). Angefangen bei den naturwissenschaftlichen Grundprinzipien der Physik, Informatik und Mathematik führte das Schulungsprogramm hin zur praxisnahen Implementierung fundamentaler Quantenalgorithmen sowohl auf der hauseigenen IAO / IAF-Simulationsumgebung als auch auf dem IBM Quantum System One in Ehningen. Der Umgang mit diesem Quantensystem und das Kennenlernen seiner Charakteristika definierten einen zentralen Aspekt der Schulungsinhalte.

Die Teilnehmenden aus Forschung und Industrie erhielten außerdem einen Einblick in eine Vielzahl von Spezialthemen, wie etwa die (hybriden) Quantenalgorithmen Deutsch-Jozsa, Bernstein-Vazirani, Grover, HHL, Shor, QFT, QPE, VQE, VQLS und QAOA (siehe

Kapitel 1.4). Diese wurden in Form von dreistündigen Webinaren bis hin zu Halbtagesseminaren vermittelt. Weiter wurden die Konzepte Fehlermitigation, effizientes Transpiling-, sowie Qiskit Pulse- oder Runtime-Methoden erläutert und diskutiert. Die Teilnehmenden konnten bei den Vertiefungen und den Anwendungsgebieten dabei stets ihre Interessen und ihr Vorwissen mit einbringen. Abschließend wurden die erlernten QC-Werkzeuge zur Realisierung konkreter Anwendungsfälle verwendet (siehe Kapitel 2).

Die Teilnehmenden lernten insbesondere die unterschiedlichen QC-Hardware- (siehe Kapitel 1.3) und Software-Ansätze (siehe Kapitel 1.4) sowie deren Vor- bzw. Nachteile für ausgewählte Anwendungsfälle kennen. Dabei standen hybride QC-Infrastrukturen und deren cloudbasierte Nutzung im Fokus. Dies geschah zum einen durch Webinare von Fraunhofer und Technologieanbietern (IBM, QuTech, Quantum Brilliance, D-Wave, QuiX, Pasqal, Fujitsu, AnaQor, JoSQuantum, HQS, Classiq, Quantinuum, AWS), als auch durch Gastbeiträge von bedeutenden nationalen Förderprojekten (European Quantum Industry Consortium, European Photonics Industry Consortium, Infineon Technologies AG, QBN, Bitkom, PlanQK, AQUAS, EniQma, AutoQML, Q.AX). Zentrales Ziel war hierbei ein proaktiver Austausch der Quantencommunity sowie eine realistische Einschätzung der aktuellen Wettbewerbslandschaft und der perspektivischen Entwicklung des QC-Ökosystems.

Jenes Schulungsprogramm wird 2023 modularisiert weitergeführt und sukzessive weiterentwickelt. Geplanter Auftakt ist dabei der 28. Februar 2023. Falls Sie Teil dieses Netzwerkes bzw. Programms werden möchten, können Sie dieses Interesse gerne über diesen [Link](#) kundtun.

Autor: Dr. Christian Tutschku (Fraunhofer IAO)

Kontakt: [christian.tutschku@iao.fraunhofer.de](mailto:christian.tutschku@iao.fraunhofer.de)

# LITERATURVERZEICHNIS

## Kapitel 1: Das Quantencomputing Ökosystem

### Einleitung

[I 1] N. Moll et al., Quantum Optimization Using Variational Algorithms on Near-term Quantum Devices, *Quantum Science and Technology* **3**(3), 030503 (2018)

[I 2] A. Wack et al., Quality, Speed, and Scale: Three Key Attributes to Measure the Performance of Near-term quantum Computers, arXiv: 2110.14108 (2021)

[I 3] C. Carugno, M. F. Dacerema, P. Cermonsi, Evaluating the Job Shop Scheduling Problem on a D-wave Quantum Annealer, *Sci. Rep.* **12**, 6539 (2022)

[I 4] W. Cruz-Santos, S. E. Venegas-Andraca, M. Lanzagorta, A QUBO Formulation of Minimum Multicut Problema Instances in Tres for D-Wave Quantum Annealers, *Sci. Rep.* **9**, 17216 (2019)

[I 5] S. Okada, M. Ohzeki, M. Terabe et al., Improving solutions by embedding larger subproblems in a D-Wave quantum annealer. *Sci. Rep.* **9**, 2098 (2018)

### Grundlagen

[G 1] acatech, Quantentechnologien, acatech HORIZONTE (2020), <https://www.acatech.de/publikation/acatech-horizonte-quantentechnologie/> (letzter Aufruf: 13.02.2023)

[G 2] E. Grumbling, M. Horowitz, Quantum Computing. Progress and Prospects, National Academies Press (2019)

[G 3] M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, Cambridge (2010)

[G 4] E. Rieffel, W. Polak, Quantum Computing. A Gentle Introduction, The MIT Press, Cambridge (2011)

[G 5] <https://www.olcf.ornl.gov/frontier/> (letzter Aufruf 13.02.2023)

[G 6] <https://www.datacenterdynamics.com/en/news/oak-ridges-exascale-frontier-system-named-worlds-most-powerful-supercomputer-on-top500/> (letzter Aufruf 13.02.2023 )

[G 7] Siemens Mobility Global, Der ICE 4. Das Rückgrat des zukünftigen DB-Fernverkehrs (2021), <https://www.mobility.siemens.com/global/de/portfolio/schiene/fahrzeuge/hochgeschwindigkeit-intercity/ice-4.html> (letzter Aufruf 6.9.2021)

[G 8] D. Willsch et al., Benchmarking Supercomputers with the Jülich Universal Quantum Computer Simulator (2020), NIC Symposium 2020

## Hardwarerealisierungen von Quantencomputern

- [H 1] Y. Chi et al., A Programmable Qudit-based Quantum processor, *Nat Commun* **13**, 1166 (2022)
- [H 2] M. H. Holzscheiter, Ion-Trap Quantum Computation, *Los Alamos Science* **27**, 264-283 (2002)
- [H 3] P. Krantz et al., A Quantum Engineer's Guide to Superconducting Qubits, *Applied Physics Reviews* **6**, 021318 (2019)
- [H 4] R. C. Ashoori, Electrons in Artificial Atoms, *Nature* **379**, 413-419 (1996)
- [H 5] M. Tinkham, Introduction to Superconductivity, Dover Publications Inc., Mineola (2004)
- [H 6] J. Bardeen, L. N. Cooper, J. R. Schrieffer, Theory of Superconductivity, *Physical Review* **108(5)**, 1175 (1957)
- [H 7] W.C. Stewart, Current-voltage Characteristics of Josephson Junctions, *Applied Physics Letters* **12(8)**, 277-280 (1968)
- [H 8] A. A. Golubov, M. Y. Kupriyanov, E. Il'ichev, The Current-phase Relation in Josephson Junctions, *Reviews of Modern Physics* **76(2)**, 411 (2004)
- [H 9] R. Barends et al., Coherent Josephson Qubit suitable for Scalable Quantum Integrated Circuits, *Physical Review Letters* **111(8)**, 080502 (2013)
- [H 10] V. E. Manucharyan et al., Fluxonium: Single Cooper-pair Circuit Free of Charge Offsets, *Science* **326(5949)**, 113-116 (2009)
- [H 11] M. Möttönen: Unimon Qubit, *Nature Communications* **13**, 6895 (2022)
- [H 12] S. Sheldon et al., Procedure for Systematically Tuning Up Cross-talk in the Cross-Resonance Gate, *Physical Review A* **93(6)**, 060302 (2016)
- [H 13] N. Sundaresan et al., Reducing Unitary and Spectator Errors in Cross Resonance with Optimized Rotary Echoes. In: *PRX Quantum* **1(2)**, 020318 (2020)
- [H 14] J. Wrachtrup, S. Y. Kilin and A. P. Nizovtsev: Quantum Computation using the <sup>13</sup>C Nuclear Spins Near the Single NV Defect Center in Diamond, *Optics and Spectroscopy* **91(3)**, 429-437 (2001)
- [H 15] A. P. Nizovtsev et al., A Quantum Computer based on NV Centers in Diamond: Optically Detected Nutations of Single Electron and Nuclear Spins, *Optics and Spectroscopy* **99(2)**, 233-244 (2005)

- [H 16] E. D. Herbschleb et al., Ultra-long Coherence Times amongst Room-temperature Solid-state Spins, *Nature Communications* **10**, 3766 (2019)
- [H 17] M.H. Abobeih et al., One-second Coherence for a Single Electron Spin coupled to a Multi-qubit Nuclear-spin Environment, *Nature Communications* **9**, 2552 (2018)
- [H 18] H. Park et al., Decoherence of Nitrogen-vacancy Spin Ensembles in a Nitrogen Electron-nuclear Spin Bath in Diamond, *npj Quantum Inf* **8**, 95 (2022)
- [H 19] N. Bar-Gill et al., Solid-state electronic spin coherence time approaching one second, *Nat Commun* **4**(1), 1743 (2013)
- [H 20] S. Barz: Quantum Computing with Photons: Introduction to the Circuit Model, the One-way Quantum Computer, and the Fundamental Principles of Photonic Experiments, *Journal of Physics B: Atomic, Molecular and Optical Physics* **48**(8), 083001 (2015)
- [H 21] Z. Zhao et al.: Experimental Demonstration of a Nondestructive Controlled-NOT Quantum Gate for Two Independent Photon Qubits. In: *Physical Review Letters* **94**(3), 030501 (2005)
- [H 22] S. Slussarenko, G.J. Pryde, Photonic Quantum Information Processing: A Concise Review, *Applied Physics Reviews* **6**, 041303 (2019)
- [H 23] Y. Li et al., Resource Costs for Fault-tolerant Linear Optical Quantum Computing, *Phys. Rev. X* **5**, 041007 (2015)
- [H 24] R. Raussendorf, H. J. Briegel, A One-way Quantum Computer, *Physical Review Letters* **86**(22), 5188 (2005)
- [H 25] A. Barenco et al., Elementary Gates for Quantum Computation, *Phys. Rev. A* **52**, 3457 (1995)
- [H 26] S. Bartolucci et al., Fusion-based Quantum Computation, arXiv:2101.09310 (2021)[H 27] B.C. Barrish, R. Weiss, LIGO and the detection of gravitational waves, *Physics Today* **52**, 10, 44 (1999)
- [H 28] J. E. Bourassa et al., Blueprint for a Scalable Photonic Fault-Tolerant Quantum Computer, *Quantum* **5**, 392 (2021)
- [H 29] Grafik entnommen von: L.S. Madsen et al. Quantum Computational Advantage with a Programmable Photonic Processor. *Nature* **606**, 75–81 (2022), unter der Creative Commons 4.0 Lizenz unter <https://creativecommons.org/licenses/by/4.0/> .
- [H 30] Li, M. et al., Femtosecond Laser Direct Writing of Integrated Photonic Quantum Chips for Generating Path-Encoded Bell States, *Micromachines* **11**, 1111 (2020)

## Quantenalgorithmen

[A 1] D. Deutsch, Quantum Theory, the Church–Turing Principle and the Universal Quantum Computer, Proc. R. Soc. Lond. A **400**, 1818, 97–117 (1985)

[A 2] D. Deutsch, R. Jozsa, Rapid Solution of Problems by Quantum Computation, Proc. R. Soc. Lond. A **439**, 1907, 553–558 (1992)

[A 3] M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2010)

[A 4] E. Rieffel, W. Polak, Quantum Computing. A Gentle Introduction, The MIT Press, Cambridge (2011)

[A 5] P. W. Shor, Algorithms for Quantum Computation: Discrete Logarithms and Factoring, Proceedings 35th Annual Symposium on Foundations of Computer Science, 124–134 (1994)

[A 6] Shor, P. W. 1997. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM J. Comput. **26**(5), 1484–1509.

[A 7] E. Grumbling, M. Horowitz, Quantum Computing. Progress and Prospects. National Academies Press (2019)

[A 8] F. Wilhelm et al., Status of Quantum Computer Development. Entwicklungsstand Quantencomputer. V1.2. (2021), [https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Quantentechnologien-und-Post-Quanten-Kryptografie/Entwicklungsstand-Quantencomputer/entwicklungsstand-quantencomputer\\_node.html](https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Quantentechnologien-und-Post-Quanten-Kryptografie/Entwicklungsstand-Quantencomputer/entwicklungsstand-quantencomputer_node.html) (letzter Aufruf 13.02.2023)

[A 9] T. Stollenwerk, Umfeldstudie Quantencomputing (2021), <https://www.dlr.de/content/de/dossiers/2021/quantencomputing.html> (letzter Aufruf 13.02.2023)

[A 10] L. K. Grover, Quantum Computers Can Search Arbitrarily Large Databases by a Single Query, Phys. Rev. Lett. **79**(23), 4709–4712 (1997)

[A 11] A. M. Childs, A. J. Landahl, P. A. Parrilo, Quantum Algorithms for the Ordered Search Problem via Semidefinite Programming, Phys. Rev. A **75**(3), 32335 (2007)

[A 12] C. H. Bennett et al., Strengths and Weaknesses of Quantum Computing, SIAM J. Comput. **26**(5), 1510–1523 (1997)

[A 13] A. Bartschi, S. Eidenbenz, Grover Mixers for QAOA: Shifting Complexity from Mixer Design to State Preparation, In IEEE International Conference on Quantum Computing and Engineering, 72–82 (2020)

[A 14] A. J. et al., Quantum Algorithm Implementations for Beginners. ACM Transactions on Quantum Computing, Volume 3, Issue 4, 1-92 (2022)



- [A 15] E. G. Rieffel, An Overview of Quantum Computing for Technology Managers, arxiv: 0804.2264v2 (2008)
- [A 16] A. W. Harrow, A. Hassidim, S. Lloyd, Quantum Algorithm for Linear Systems of Equations, Phys. Rev. Lett. **103**(15), 150502 (2009)
- [A 17] A. Ambainis, Variable Time Amplitude Amplification and a Faster Quantum Algorithm for Solving Systems of Linear Equations, arxiv: 1010.4458v2 (2010)
- [A 18] A. M. Childs, R. Kothari, R. D. Somma, Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision, SIAM J. Comput. **46**(6), 1920–1950 (2017)
- [A 19] B. D. Clader, B. C. Jacobs, C. R. Sprouse, Preconditioned Quantum Linear System Algorithm, Phys. Rev. Lett. **110**(25), 250504 (2013)
- [A 20] L. Wossnig, Z. Zhao, A. Prakash, Quantum Linear System Algorithm for Dense Matrices, Phys. Rev. Lett. **120**(5), 50502 (2018)
- [A 21] D. Dervovic et al., Quantum Linear Systems Algorithms: a Primer, arxiv: 1802.08227v1 (2018)
- [A 22] A. Scherer, et al., Concrete Resource Analysis of the Quantum Linear System Algorithm Used to Compute the Electromagnetic Scattering Cross Section of a 2D Target, Quantum Inf Process **16**(3), 150502 (2017)
- [A 23] K. Bharti et al., Noisy Intermediate-Scale Quantum (NISQ) Algorithms, Rev. Mod. Phys. **94**(1), 15004 (2022)
- [A 24] M. Cerezo et al., Variational Quantum Algorithms. Nat Rev Phys, 1–20 (2021)
- [A 25] S. Endo et al., Hybrid Quantum-Classical Algorithms and Quantum Error Mitigation, J. Phys. Soc. Jpn. **90**(3), 32001 (2021)
- [A 26] N. Moll et al., Quantum Optimization Using Variational Algorithms on Near-Term Quantum Devices, Quantum Sci. Technol. **3**(3), 30503 (2018)
- [A 27] G. Nannicini, Performance of Hybrid Quantum-Classical Variational Heuristics for Combinatorial Optimization, Physical review. E **99**, 1-1, 13304 (2019)
- [A 28] Y. Cao et al., Quantum Chemistry in the Age of Quantum Computing. Chemical Reviews **119**(19), 10856–10915 (2019)
- [A 29] E. Farhi, J. Goldstone, S. Gutmann, A Quantum Approximate Optimization Algorithm, arXiv:1411.4028 (2014)
- [A 30] M. Anthony et al., Quadratic Reformulations of Nonlinear Binary Optimization Problems, Math. Program. **162**, 1-2

- [A 31] F. Glover, G. Kochenberger, Y. Du. Quantum Bridge Analytics I: A Tutorial on Formulating and Using QUBO Models, *4OR-Q J Oper Res* **17**(4), 335–371 (2019)
- [A 32] S. Hadfield et al., Quantum Approximate Optimization with Hard and Soft Constraints, *Proceedings of the Second International Workshop on Post Moores Era Supercomputing, ACM*, 15–21 (2017)
- [A 33] G. Kochenberger et al., The Unconstrained Binary Quadratic Programming Problem: A Survey, *J Comb Optim* **28**(1), 58–81 (2014)
- [A 34] A. Lucas, Ising Formulations of Many NP Problems, *Front. Physics* **2** (2014)
- [A 35] R. Shaydulin, I. Safro, J. Larson, Multistart Methods for Quantum Approximate optimization, *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, 1–8 (2019)
- [A 36] S. Hadfield et al., From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz, *Algorithms* **12**(2), 34 (2019)
- [A 37] D. J. Egger, J. Marecek, S. Woerner, Warm-Starting Quantum Optimization, *Quantum* **5**, 479 (2021)
- [A 38] M. Streif, M. Leib, Comparison of QAOA with Quantum and Simulated Annealing, arxiv: 1901.01903v1 (2019).
- [A 39] L. Page et al., The PageRank Citation Ranking: Bringing Order to the Web. *Stanford InfoLab* (1999)
- [A 40] Y. Saad, *Numerical Methods for Large Eigenvalue Problems, Classics in applied mathematics* 66, Society for Industrial and Applied Mathematics, Philadelphia (2011)
- [A 41] S. McArdle et al., Quantum Computational Chemistry, *Rev. Mod. Phys.* **92**(1), 15003 (2020)
- [A 42] J. McClean et al., The Theory of Variational Hybrid Quantum-Classical Algorithms. *New J. Phys.* **18**(2), 23023 (2016)
- [A 43] A. Peruzzo et al., A Variational Eigenvalue Solver on a Photonic Quantum Processor. *Nat Commun* **5**(1), 4213 (2014)
- [A 44] D. Coppersmith, An Approximate Fourier Transform Useful in Quantum Factoring, arxiv: quant-ph/0201067 (2002)
- [A 45] S. Jordan, Quantum Algorithm Zoo, <https://quantumalgorithmzoo.org/> (letzter Aufruf 13.02.2023)
- [A 46] S. Aaronson, Read the Fine Print, *Nature Phys* **11**, 4, 291–293 (2015)
- [A 47] R. Babbush et al., Focus Beyond Quadratic Speedups for Error-Corrected Quantum Advantage. *PRX Quantum* **2**(1), 10103 (2021)

## Exkurs Quantenfehlerkorrektur

[E 1] M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, Cambridge (2010)

[E 2] D. Gottesman, Theory of Fault-Tolerant Quantum Computation, Phys. Rev. A **57**, 1, 127–137 (1998)

[E 3] P. W. Shor, Fault-Tolerant Quantum Computation, Proceedings of 37th Conference on Foundations of Computer Science, 56–65 (1996)

[E 4] A. G. Fowler, A. M. Stephens, P. Groszkowski, High-Threshold Universal Quantum Computation on the Surface Code. Phys. Rev. A **80**(5), 52312 (2009)

[E 5] E. T. Campbell, B. M. Terhal, C. Vuillot, Roads Towards Fault-Tolerant Universal Quantum Computation, Nature **549**, 172–179 (2017)

[E 6] J. Preskill, Quantum Computing in the NISQ Era and Beyond, Quantum **2**, 79 (2018)

[E 7] F. B. Maciejewski, Z. Zimborás, M. Oszmaniec, Mitigation of Readout Noise in Near-Term Quantum Devices by Classical Post-Processing Based on Detector Tomography. Quantum **4**, 257 (2020)

[E 8] A. He et al., Zero-Noise Extrapolation for Quantum-Gate Error Mitigation with Identity Insertions, Phys. Rev. A **102**(1), 12426 (2020).

[E 9] M. Beisel et al., Configurable Readout Error Mitigation in Quantum Workflows, Electronics **11**, 2983 (2022)

## Kapitel 2: Anwendungsfälle

### Fokus-Anwendung 1

[UC 1] <https://www.lama.zone/>

[UC 2] A. Sturm, SEQUOIA DEMONSTRATOR Energy Use Case, notebook\_1, Zugriff via [Use Cases – SEQUOIA \(fraunhofer.de\)](#)

[UC 3] qiskit.algorithms.minimum\_eigen\_solvers.qaoa.QAOA, Version qiskit 0.39.2, qiskit-terra 0.22.2

[UC 4] qiskit.circuit.library.n\_local.qaoa\_ansatz.QAOAAnsatz, Version qiskit 0.39.2, qiskit-terra 0.22.2

[UC 5] A. Sturm, SEQUOIA DEMONSTRATOR Energy Use Case, notebook\_2, Zugriff via [Use Cases – SEQUOIA \(fraunhofer.de\)](#)

[UC 6] A. Sturm, SEQUOIA DEMONSTRATOR Energy Use Case, notebook\_3, Zugriff via [Use Cases – SEQUOIA \(fraunhofer.de\)](#)

[UC 7] A. Sturm, SEQUOIA DEMONSTRATOR Energy Use Case, notebook\_4, Zugriff via [Use Cases – SEQUOIA \(fraunhofer.de\)](#)

[UC 8] Sturm, A. 2023. Theory and Implementation of the Quantum Approximate Optimization Algorithm: A Comprehensible Introduction and Case Study Using Qiskit and IBM Quantum Computers. arXiv:2301.09535v1

[UC 9] A. Sturm, N. Schillo, SEQUOIA DEMONSTRATOR Energy Use Case, notebook\_5, Zugriff via [Use Cases – SEQUOIA \(fraunhofer.de\)](#)

### Fokus-Anwendung 2

[UC 10] A. W. Harrow, A. Hassidim, S. Lloyd, Quantum Algorithm for Linear Systems of Equations, Phys. Rev. Lett. **103**, 150502 (2009)

[UC 11] C. Bravo-Prieto et al., Variational Quantum Linear Solver, arXiv:1909.05820v2 (2019)

### Fokus-Anwendung 3

[UC 12] J. Zhang, G. Lo Bianco, J. C. Beck, Solving Job-Shop Scheduling Problems with QUBO-Based Specialized Hardware, Proceedings of the International Conference on Automated Planning and Scheduling, **32**(1), 404-412 (2022)

[UC 13] K. Kurowski et al., Hybrid Quantum Annealing Heuristic Method for Solving Job Shop Scheduling Problem, Computational Science – ICCS 2020. Lecture Notes in Computer Science **12142**, 502-515, Springer, Cham (2022)

### Fokus-Anwendung 4

[UC 14] E. Fahri, J. Goldstone, S. Gutmann, Sam, A Quantum Approximate Optimization Algorithm, arXiv:1411.4028 (2014)

[UC 15] J. Jylänki, A Thousand Ways to Pack the Bin – A Practical Approach to Two-dimensional Rectangle Bin Packing (2010),  
<http://pds25.egloos.com/pds/201504/21/98/RectangleBinPack.pdf>  
(letzter Aufruf 13.02.2023)

[UC 16] A. Lucas, Ising Formulations of Many NP Problems, Frontiers in Physics **2**, 5 (2014)

[UC 17] J. F. Oliveira, A. M. Gomes, J. S. Ferreira, TOPOS – A New Constructive Algorithm for Nesting Problems, OR-Spektrum **22**(2), 263-284 (2000)

[UC 18] K. Terada et al., An Ising Model Mapping to Solve Rectangle Packing Problem, 2018 International Symposium on VLSI Design, Automation and Test, 1-4 (2018)

### Fokus-Anwendung 5

[UC 19] P. Vansteenwegen, W. Souffriau, D. Van Oudheusden, The Orienteering Problem: A Survey, European Journal of Operational Research **209**(1), 1-10 (2011)

[UC 20] S. Hadfield et al., From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz, Algorithms **12**(2), 34 (2019)

[UC 21] P. E. Hart, N. J. Nilsson, B. Raphael, A Formal Basis for the Heuristic Determination of Minimum Cost Paths, IEEE Transactions on Systems Science and Cybernetics **4**(2), 100-107 (1968)

[UC 22] S. Garhofer, SEQUOIA DEMONSTRATOR Bertrandt Use Case, notebook\_2, Zugriff via [Use Cases – SEQUOIA \(fraunhofer.de\)](#)

### Kapitel 3: Software Engineering

[S 1] D. Pfahl, ACM SIGSOFT Software Engineering Notes, **46**(2), 30–32 (2021)

[S 2] R. Perez-Castillo et al., Proceedings of the 1st International Workshop on Software Engineering & Technology, Denver – Broomfield (2020)

[S 3] R. Perez-Castillo et al., Short Papers Proceedings of the 2nd International Workshop on Software Engineering & Technology, Online Conference (2021)

[S 4] J. Barzen et al., 1st Workshop on Quantum Software Architecture (QSA), IEEE 18th International Conference on Software Architecture Companion (ICSA-C), 94-94 (2021)

[S 5] Proceedings of the 2022 IEEE International Conference on Quantum Software (QSW), Barcelona (2022)

[S 6] M. Weigold et al., Patterns for Hybrid Quantum Algorithms. Proceedings of the 15th Symposium and Summer School on Service-Oriented Computing, Communications in Computer and Information Science **1429**, 34-51, Springer, Cham (2021)

[S 7] M. Scheerer, J. Klamroth, O. Denninger, Engineering Reliable Hybrid Quantum Software: An Architectural-driven Approach, Short Papers Proceedings of the 2nd International Workshop on Software Engineering & Technology (Q-SET 2021), Virtual Conference (2021)

[S 8] L. Westfall, A. Leider, . Teaching Quantum Computing, Proceedings of the Future Technologies Conference (FTC) 2018, Adv. Intell. Syst. Comput. **881**, Springer, Cham (2019)

[S 9] J. R. Wootton et al., Teaching quantum computing with an interactive textbook, 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), 385-391(2021)

[S 10] N. Kiefl, G. Hagel, Software Engineering Education of Classical Computing vs. Quantum Computing, A Competency-Centric Approach, Proceedings of the 4th European Conference on Software Engineering Education (ECSEE '20), Association for Computing Machinery, New York, 27–31 (2020)

S 11] N. Kiefl, G. Hagel, Software Engineering Education of Classical Computing vs. Quantum Computing, A Competency-Centric Approach, Proceedings of the 4th European Conference on Software Engineering Education (ECSEE '20), Association for Computing Machinery, New York, 27–31 (2020)

[S 12] L. Gatti, R. Sotelo, Quantum Computing for Undergraduate Engineering Students: Report of an Experience, 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), 397-401 (2021)

[S 13] G. Carrascal, A.A. del Barrio, G. Botella, First Experiences of Teaching Quantum Computing. J Supercomput **77**, 2770–2799 (2021)

- [S 14] A. Miransky, L. Zhang, J. Doliskani, On Testing and Debugging Quantum Software, arXiv:2103.09172, (2021)
- [S 15] J. Patrzyk et al., Towards A Novel Environment for Simulation of Quantum Computing, Computer Science, 16(1), 103 (2015)
- [S 16] M. Salm et al., The NISQ Analyzer: Automating the Selection of Quantum Computers for Quantum Algorithms, Service-Oriented Computing SummerSOC 2020, Communications in Computer and Information Science, 1310, Springer, Cham (2020)
- [S 17] M. Weigold et al., Encoding Patterns for Quantum Algorithms, IET Quant. Comm. 2(4), 141–152 (2021)
- [S 18] E. Farhi, J. Goldstone, S. Gutmann. A Quantum Approximate Optimization Algorithm, arXiv:1411.4028 (2014)
- [S 19] S. Hadfield et al., From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz, arXiv:1709.03489, (2019)
- [S 20] B. Weder, Integrating Quantum Computing into Workflow Modeling and Execution, Proceedings of the 13th IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2020), IEEE Computer Society (2020)
- [S 21] A. W. Cross et al., Validating Quantum Computers Using Randomized Model Circuits, Physical Review A **100**, 032328 (2019)
- [S 22] B. Weder, The Quantum Software Lifecycle, Proceedings of the 1st ACM SIGSOFT International Workshop on Architectures and Paradigms for Engineering Quantum Software (APEQS 2020), Association for Computing Machinery, New York, 2–9, (2020)
- [S 23] N. Tcholtchev et al., Quantum DevOps: Towards Reliable and Applicable NISQ Quantum Computing, 2020 IEEE Globecom Workshops (2020)

## ABBILDUNGSVERZEICHNIS

Abbildung 1: Vergleich der möglichen Zustände eines Bits und eines Qubits. Die Darstellung beim Qubit ist aus Gründen der Niederschwelligkeit vorerst vereinfacht, eine präzise Visualisierung wird in Abschnitt 1.2.2 besprochen. ...	9
Abbildung 2: Darstellung des Qubits $ \psi\rangle$ auf der Bloch-Sphäre.....	10
Abbildung 3: Die Messung eines Qubits kann nur 0 oder 1 zum Ergebnis haben. Der Breitengradwinkel $\theta$ codiert mit welcher Wahrscheinlichkeit welches Ergebnis auftreten wird. Je nördlicher sich das Qubit $ \psi\rangle$ befindet desto wahrscheinlicher ist 0, je südlicher es sich befindet desto wahrscheinlicher wird 1. Der Längengradwinkel $\Phi$ spielt für die Messung keine Rolle. ....	11
Abbildung 4: Beispiele (a), (b) und (c) visualisiert auf der Bloch-Sphäre. ....	13
Abbildung 5: Durch die Messung kollabiert der Superpositionszustand von $ \psi\rangle$ . ....	14
Abbildung 6: Stabmagnet (Kompassnadel) im homogenen Feld eines Hufeisenmagneten. Die magnetischen Nord- und Südpole sind mit N (rot) und S (grün) gekennzeichnet. Die Teilabbildungen a) bzw. b) definieren die energetisch niedrigsten und höchsten Ausrichtungen des Stabmagnetes im homogenen externen Feld: a) Anti-parallele Ausrichtung der Magnetfeldlinien (Grundzustand) bzw. b) parallele Ausrichtung der Magnetfeldlinien (angeregter Zustand). ....	18
Abbildung 7: Energetischer Grund- $ 0\rangle$ und angeregter Zustand $ 1\rangle$ eines Quantensystems, die mittels Emission- bzw. Absorption eines Energiequants (rote gewellte Linie) ineinander überführt werden können. Derartige Übergänge werden auf einem Quantencomputer durch sogenannte Gates induziert. Beispiel: Vom Grundzustand $ 0\rangle$ in den angeregten Zustand $ 1\rangle$ eines Quantensystems gelangt man durch das Anwenden eines X-Gates.....	19
Abbildung 8: Kugel- (s) und Hantelatomorbitale (p) sowie zugehörige Energieniveaus eines Elektrons in einem Atom (strukturell für die Hauptquantenzahlen $n = 1$ und $n = 2$ ). Übergänge im Spektrum können mittels Emission bzw. Absorption von Energiequanten, z.B. in Form von Licht (elektromagnetischen Wellen) erzeugt werden. Bei Energieniveaus gleicher Energie spricht man von Entartung. Die p-Orbitale unterscheiden sich in der Quantenzahl für den Bahndrehimpuls des Elektrons. Im einfachsten Modell sind diese entartet, wobei die Entartung aufgespalten wird, wenn weitere Effekte wie Spin-Bahn Kopplung oder eine Wechselwirkung mit einem äußeren elektrischen oder magnetischen Feld vorliegen.....	20
Abbildung 9: Aufbau einer linearen Paul Falle – Das Grundprinzip eines Ionen-basierten Quantencomputers. Geladene Atome oder Moleküle werden mittels oszillierender elektromagnetischer Felder eingefangen. Weitere Erklärungen sind dem Fließtext zu entnehmen. Die Graphik ist angelehnt an Referenz [H 2]. ....	22
Abbildung 10: a) Elektronischer Schaltplan eines LC-Schwingkreises; b) zugehöriges quantisiertes Energiespektrum (sogenannter harmonischer Oszillator); c) einfaches Qubit, realisiert durch eine parallel geschaltete Induktivität L, einen Josephson-Kontakt mit Energie $EJ$ und eine Kapazität C; d) zugehöriges Energiespektrum des Transmon-Qubits. Der bedeutende Unterschied zwischen dem Energiespektrum des supraleitenden LC-Schwingkreises in (b) und dem des Transmon-Qubits in (d) liegt in der Anharmonizität. Während in (b) die diskreten Energieniveaus äquidistant zueinander liegen, vergrößert sich diese Differenz mit zunehmender Energie beim Transmon-Qubit. Um den Unterraum aller Zustände auf das $ 0\rangle$ und $ 1\rangle$ Subsystem zu beschränken, muss diese Anharmonizität erzeugt werden. Anderenfalls würde eine Anregung von $ 0\rangle$ nach $ 1\rangle$ gleichzeitig auch $ 1\rangle$ nach $ 2\rangle$ , $ 2\rangle$ nach $ 3\rangle$ usw. induzieren. ....	25
Abbildung 11: Schematische Veranschaulichung der Funktionsweise eines Josephson-Kontakts. Eine dünne (wenige Nanometer breite) Isolator-Barriere aus nichtleitendem Material wird in einen Supraleiter eingefügt, wodurch zwei Supraleiter entstehen. Deren Cooper-Paare, die für die Supraleitung verantwortlich sind, können jeweils durch die Wellenfunktionen $\Psi_{A/B}$	



beschrieben werden. Ist der Isolator hinreichend dünn kommt es zum quantenmechanischen Phänomen des »Tunnelns«, wobei sich die jeweiligen Wellenfunktionen in die Bereiche des anderen Supraleiters ausdehnen können. Es resultiert eine nicht-lineare Strom-Spannungscharakteristik, die für das Transmon-Qubit verwendet wird. ....	26
Abbildung 12: Illustration eines supraleitenden Quantenchips mit vier Transmon Qubits. Zu erkennen sind die Resonatoren, die an die Transmons gekoppelt sind, sowie die Mikrowellenleiter, mit denen die Transmons über passende Pulssequenzen initialisiert, kontrolliert und ausgelesen werden ©IBM. ....	27
Abbildung 13: Reines Diamantkristallgitter (links) und Präsenz eines Stickstoffvakanz-(NV-) Zentrums (rechts) in der ansonsten geordneten Struktur der Kohlenstoffatome (schwarze Kugeln). Weitere Erklärungen sind dem Fließtext zu entnehmen. ....	29
Abbildung 14: Illustration eines NV-Zentrums in einer lokalen Ansammlung von hyperfein-gekoppelten $^{13}\text{C}$ Kernspins. Der Elektronenspin koppelt an die Nukleare spins ( $^{13}\text{C}$ und N), die wiederum als Qubits genutzt werden. Die Kernspins besitzen aufgrund der lokalen Felder im Kristall ihre jeweils eigene Resonanzfrequenz. Üblicherweise würden die Kernspins den Elektronenspin stören, jedoch kann über Spin-Echos, der Spin bis zu einer Größenordnung von 1 Sekunde geschützt werden [H19]. ....	32
Abbildung 15: Schematische Darstellung für die Skalierung von NV-Registern. Detaillierte Informationen sind dem Fließtext zu entnehmen. ....	33
Abbildung 16: (Links) Die Basiszustände eines photonischen Qubits ( $ 0\rangle$ und $ 1\rangle$ ) können über die Polarisationszustände der Lichtwelle $ H\rangle$ (horizontal) und $ V\rangle$ (vertikal) kodiert werden. Analog zur Bloch-Kugel für ein Qubit existiert für die Polarisationszustände der Lichtwelle die sogenannte Poincaré-Kugel (rechts). Um eine Superposition der horizontalen und vertikalen Polarisation zu erzeugen, kann eine sogenannte Lambda-halbe Wellenplatte genutzt werden. ....	35
Abbildung 17: Clusterzustände sind eine spezielle Art von Ressourcenzustand, die im messungsbasierten Quantencomputing Verwendung finden. Die einzelnen Qubits bilden die Eckpunkte „vertices“ eines Graphen. Die Qubits werden mit controlled-Z (CZ) Gattern untereinander verschränkt. Mithilfe von lokalen Messungen können anschließend beliebige unitäre Gatter realisiert werden. ....	36
Abbildung 18: Schematische Darstellung eines photonischen Quantenprozessors der Firma Xanadu. Für die detaillierte Beschreibung der Funktionsweise des Prozessors sei auf Referenz [H 29] verwiesen aus der die Grafik entnommen wurde. Man präpariert einen Quantenzustand über Boson Sampling, welcher dann in der QPU verarbeitet wird. ....	38
Abbildung 19 Schematische Darstellung eines photonischen Quantenchips, der aus einem H- und einem CNOT-Gatter besteht, um pfadkodierte Bell-Zustände zu erzeugen. Der Richtkoppler im rot gestrichelten Kasten stellt ein Hadamard-Gatter dar, der Rest des Chips ein CNOT-Gatter. Die mit $C_0$ und $C_1$ markierten Wellenleiter stellen das Control-Qubit dar, während $T_0$ und $T_1$ für das Target-Qubit stehen. Die übrigen Wellenleiter sind Hilfs-Qubits (Ancillas), welche für die Umsetzung der Gatteroperationen auf diesem Chip benötigt werden. Jene Grafik wurde aus Referenz [H 30] übernommen. ....	38
Abbildung 20: Der Grover-Algorithmus benötigt ein Orakel mit folgender Funktionalität: Als Eingabe erhält es einen Eintrag aus der Datenbank und als Ausgabe liefert es ein binäres Resultat: 1, falls es sich um einen gesuchten Eintrag aus der Datenbank gehandelt hat und 0, falls nicht. ....	42
Abbildung 21: Anwendungsgebiete variationellener Quantenalgorithmien, siehe [A 28] ...	45
Abbildung 22: Funktionsweise eines variationellen Quantenalgorithmus: Für ein Satz an Parametern $\theta_1, \dots, \theta_p$ liefert der Quantencomputer die Auswertung $F(\theta_1, \dots, \theta_p)$ der Zielfunktion. Auf dem klassischen Computer wird daraus der nächste zu überprüfende Parametersatz $\theta_1, \dots, \theta_p$ berechnet und wieder an den Quantencomputer übergeben. Dies wird so lange iteriert bis ein optimaler Satz an Parametern erzeugt wurde oder andere Abbruchkriterien eintreten. ....	46

Abbildung 23: Proof-of-Concept-Modell zum Fokus-Anwendungsfall »Optimierung von Ladeplänen«. (Beispiel für abstrakte Beschriftung: 1 Zeitslot = 1 Stunde, 1 Ladelevel = 5 kW).....	55
Abbildung 24: Vier Beispiele für Proof-of-Concept Modelle. Die Problemstellung besteht immer aus einem E-Auto, einer Ladesäule, vier verfügbare Zeitslots, vier mögliche Ladelevel und einer zu ladenden Energie. Die Beispiele unterscheiden sich lediglich in Ankunfts- und Abfahrtszeiten. Je kürzer das E-Auto an der Ladesäule ist, desto flacher werden später die QAOA-Schaltkreise. ....	57
Abbildung 25: Optimierungslandschaft zu Beispiel1p3 (siehe Abbildung links) für QAOA mit $p=1$ , d.h. mit zwei freien Parametern $\beta$ und $\gamma$ . Die Farbskala gibt den Wert der Kostenfunktion an. Ziel des klassischen Optimierungsteil in QAOA ist es Parameter zu finden, für welche die Kostenfunktion minimal ist. ....	57
Abbildung 26: Ergebnis des klassischen Optimierers COBYLA für 50 zufällig gewählte Startpunkte $\beta_0 \in [0, \pi]$ und $\gamma_0 \in [0, 2\pi]$ mit einer exakten State-Vector-Simulation. Der Parameter penalty auf der x-Achse geht in der Transformation des Ausgangsproblems zu einem QUBO ein. ....	58
Abbildung 27: Qualität der vom IBM Quantum System One in Ehningen erhaltenen Lösung im Vergleich zu einer exakten Simulation (gemessen als fidelity, wobei 0 sehr schlecht und 1 sehr gut ist). Von links nach rechts sind drei verschiedene Beispiele, wobei das E-Auto links am kürzesten an der Ladesäule ist und rechts am längsten, siehe Abbildung 24. Die Tiefe der QAOA-Schaltkreise nimmt dadurch von links nach rechts zu. Es wurden 75 verschiedene Transpilationen durchgeführt, daher die variierende CNOT Gate Anzahl. Beschriftung der x-Achse: std=standard Transpilation mit Qiskit-Transpiler und optimization_level=3, dd=standard Transpilation und zusätzlich Dynamical Decoupling, xx_mit=xx und zusätzlich Auslesefehlerkorrektur. ....	59
Abbildung 28: Qualität der vom IBM Quantum System One in Ehningen erhaltenen Lösung im Vergleich zu einer exakten Simulation des gleichen Beispiels an verschiedenen Tagen. Zur Bedeutung dd und dd_mit und zum Begriff fidelity, siehe Abbildung 27. ....	59
Abbildung 29: Schematisches Diskretisieren und Flussprofil einer zweidimensionalen Strömung zu einem gegebenen Zeitpunkt $t = t_0$ . ....	60
Abbildung 30: Schaltkreistiefe und Qubitanzahl des VQLS- vs. HHL-Algorithmus in Abhängigkeit der Größe des zu lösenden linearen Gleichungssystems. ....	61
Abbildung 31: Iteratives Lösen des linearen Gleichungssystems $Ax = b$ mittels der globalen Kostenfunktion $CG$ und lokalen Kostenfunktion $CL$ mittels QASM Simulator. Die Definition der einzelnen Variablen und Operatoren sind [QCFD_2] zu entnehmen. ....	62
Abbildung 34: Konvergenzkurve (links) und Resultat (rechts) des VQLS-Algorithmus auf dem IBM Quantum System One in Ehningen. Als Optimierer kam SPSA zum Einsatz, als Matrix $A$ die 4 x 4 Identitätsmatrix und als rechte Seite $b$ der Vektor $[0.2, 0.8, 0.8, 0.2]$ . In der Mitte ist die exakte Lösung dargestellt. ....	64
Abbildung 35: Situation wie in Abbildung 30, aber als Matrix $A$ kam die 8 x 8 Identitätsmatrix zum Einsatz und als rechte Seite $b$ der Vektor $0.2, 0.4, 0.6, 0.8, 0.8, 0.6, 0.4, 0.2$ . ....	64
Abbildung 36: Situation wie in Abbildung 31 aber als Matrix $A$ wurde eine tridiagonale Matrix verwendet sowie als rechte Seite $b$ der Vektor $[1, 1, 1, 1]$ . ....	64
Abbildung 37: Exemplarische Lösung für ein Job Shop Scheduling Problem .....	66
Abbildung 38: Exemplarische Wahrscheinlichkeitsverteilung der Lösungen bei Nutzung des QAOA-Ansatzes auf dem IBM System One (optimale Lösung rot markiert) .....	67
Abbildung 39: Beispielhafte Darstellung typischer Teile, die in unseren Berechnungen platziert werden (links). Resultat eines erfolgreichen Packvorgangs (rechts)...70	70
Abbildung 40: Exemplarische Resultate des quanten-hybriden Packvorgangs verschiedener Beispiele – gerechnet auf dem IBM Quantum System One in Ehningen. ....	73
Abbildung 41: Verteilung der Häufigkeit der gemessenen Bitstrings direkt nach der Ausführung auf dem QC in Ehningen (links) und Verteilung der Häufigkeit der Strings, nachdem ungültige Touren herausgefiltert wurden.....	75

Abbildung 42: Darstellung einer optimalen TSP-Tour (links) und die approximative Lösung, die durch Problemaufteilung und Lösung auf dem Quantenrechner entstanden ist. ....	76
Abbildung 43: Ein Vehicle Routing Problem mit 9 Lieferungen als Set Partitioning Problem und die beispielhafte Lösung eines der drei dadurch entstehenden Pickup and Delivery Probleme. ....	77
Abbildung 44: Eine mögliche Lösung eines Pickup and Delivery Problems eines LKW mit 8 Lieferungen und seine Enkodierung als Permutationsmatrix $x_{i,j}$ . ....	79
Abbildung 45: Eine mögliche Lösung eines Vehicle Routing Problems mit zwei LKWs und 8 Lieferungen und seine Enkodierung als zerlegte Permutationsmatrix. ....	79
Abbildung 46: Wahrscheinlichkeitsverteilung der QAOA-Lösung eines Vehicle Routing Problems mit einem LKW und 3 Lieferungen gerechnet mit 1000 shots auf dem IBM Quantum System One in Ehningen. Die Optimale Lösung ist gekennzeichnet. ....	80
Abbildung 47: Wahrscheinlichkeitsverteilung der QAOA-Lösung eines Vehicle Routing Problems mit einem LKW und 4 Lieferungen gerechnet mit 1000 shots mit dem QASM-Simulator. Die beiden optimalen Lösungen sind gekennzeichnet. ....	80
Abbildung 48: Abstraktionsebenen im Software-Engineering .....	83
Abbildung 49: Phasen der Softwareentwicklung .....	85
Abbildung 50: Abstraktionsebenen der Programmierung von Quantencomputern. Entsprechend der Abstraktionsebene ändern sich die Anforderungsprofile für die Programmierung. Für die Erstellung von Schaltkreisen ist ein detailliertes Verständnis der physikalischen Eigenschaften der einzelnen Quantengatter sowie der durch die Gatter realisierten Effekte auf Qubits notwendig. Im Gegensatz dazu kann auf Ebene von Anwendungsbaukästen mit wesentlich weniger Detailkenntnis der physikalischen Grundlagen programmiert werden. ....	88
Abbildung 51: Schematische Darstellung der Verwendung des Werkzeugkastens .....	93
Abbildung 52: Dynamische Programmierung.....	99
Abbildung 53: Wissenstransfer- und Öffentlichkeitsarbeit im SEQUOIA Projekt. Die konkrete Liste der zugrundeliegenden Veranstaltungen ist unter [WT1] zu finden .....	101

## TABELLENVERZEICHNIS

Tabelle 1: Beispiele für unterschiedliche Qubit-Zustände mit zugehörigen Wahrscheinlichkeiten bei einer Messung das Ergebnis 0 oder 1 zu erhalten. ...	12
Tabelle 2: Beispiele aus Tabelle 1 in Koordinaten der Bloch-Sphäre. ....	13
Tabelle 3: Anzahl möglicher Zustände für Qubit-Systeme und benötigter Speicher auf einem klassischen Computer. ....	16

## INFOBOXENVERZEICHNIS

Infobox 1: Quantenvolumen und CLOPS .....	2
Infobox 2: Komplexe Zahlen .....	9
Infobox 3: Globale Phase .....	10
Infobox 4: Qubits messen: Beispiele .....	12
Infobox 5: Qubits messen: Beispiele (Bloch-Sphäre-Version) .....	13
Infobox 6: Analogie: Münzwurf .....	14
Infobox 7: Quantenparallelismus am Sudoku-Beispiel.....	17
Infobox 9: Spin .....	30
Infobox 10: Hyperfeinstruktur .....	31
Infobox 11: Polarisierung .....	35
Infobox 12: Gequetschtes Licht .....	37
Infobox 13: O-Notation .....	40
Infobox 14: Drei Qubit Bitflip Code .....	50
Infobox 15: Fehlermitigationsverfahren .....	52
Infobox 16: Algorithmus zum Lösen des zweidimensionalen Zuschnittproblems .....	72