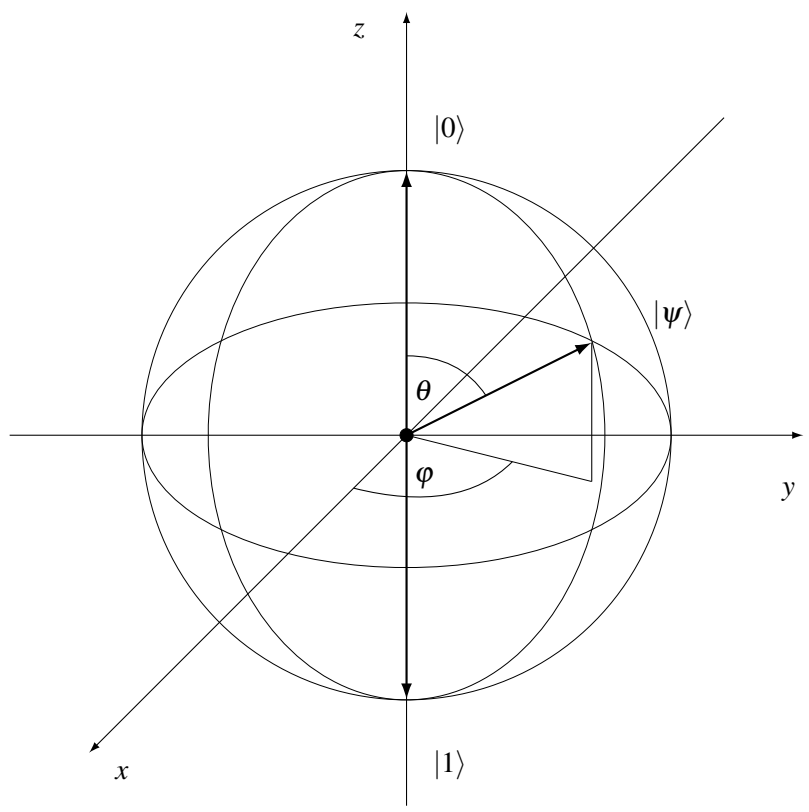
Mathematik
und
Quantum Computing

Burkhard Lenze



Logos Verlag Berlin

λογος

1 Allgemeine Einführung

Das Gebiet des sogenannten **Quantum Computing** (wir verwenden hier und im Folgenden diesen englischen Begriff, da sich die Bezeichnung **Quantenprogrammierung** nicht durchgesetzt hat) ist entstanden aus der intensiven Beschäftigung mit Fragen aus dem Bereich der **Quantenmechanik**. Ziel wird es sein, einige wesentliche Aspekte und Konzepte aus dem Computing Umfeld herauszuarbeiten und zu verstehen, wobei unser primäres Hilfsmittel die Mathematik über speziellen komplexen Vektorräumen und ihren Tensorprodukten sein wird. Insgesamt handelt es sich um ein ausgesprochen anspruchsvolles Thema, so dass wir auf keinen Fall für uns in Anspruch nehmen, es erschöpfend zu behandeln. Zum eigenen Einstieg, unabhängig von dieser Einführung, bietet es sich an, z.B. mit den Büchern von Homeister [19], Lipton und Regan [22] oder Yanofsky und Mannucci [41] zu beginnen, dann zum Quasi-Standardwerk von Nielsen und Chuang [25] zu wechseln und zum Abdecken aktueller Entwicklungen die sehr ausführliche Datenbank der Cornell University unter https://arxiv.org/list/quant-ph/recent zu studieren. Zunächst aber sollen zur Einstimmung in die Quantenwelt und ihrer Optionen hinsichtlich eines neuen Computing Paradigmas in lockerer Reihenfolge und ohne weitere Kommentare einige Zitate (Details siehe https://de.wikiquote.org/wiki/Quantenphysik) genannt werden sowie ein (sicher unvollständiger) grober Abriss der historischen Entwicklung gegeben werden (Details siehe https://en.wikipedia.org/wiki/Timeline_of_quantum_computing).

- **Niels Bohr:** *Denn wenn man nicht zunächst über die Quantentheorie entsetzt ist, kann man sie doch unmöglich verstanden haben.*
- **Albert Einstein:** *Die Quantenmechanik ist sehr Achtung gebietend. Aber eine innere Stimme sagt mir, dass das noch nicht der wahre Jakob ist. Die Theorie liefert viel, aber dem Geheimnis des Alten bringt sie uns kaum näher. Jedenfalls bin ich überzeugt, dass der Alte nicht würfelt.*
- **Max Born:** *Die Quanten sind doch eine hoffnungslose Schweinerei!*
- **Richard Feynman:** *I couldn't reduce it to the freshman level. That means we really don't understand it.*

Nach diesen schon recht vielsagenden Zitaten einiger Giganten aus dem Bereich der Quantenmechanik folgt ein kurzer historischer Überblick unter Einbeziehung der Entwicklungen im uns primär interessierenden Bereich des Quantum Computings:

- **1900:** Planckscher Wirkungsquantum.
  Die Energieabgabe eines schwarzen Körpers ist bei zunehmender Frequenz klassisch nicht erklärbar (Ultraviolettkatastrophe). Planck findet im Jahr 1900 eine zunächst rein mathematische Lösung zur befriedigenden Beschreibung: Abgegebene Strahlungsenergie ist nur in Vielfachen einer grundlegenden Konstante möglich, die später den Namen **Planckscher Wirkungsquantum** bekommt.
- **1905:** Erklärung des photoelektrischen Effekts durch Einstein.
  Einstein erkennt als erster, dass die **Quantisierung der Energie** in Plancks Formel nicht nur ein bloßer Rechentrick ist, sondern ein wirklich in der Natur vorkommendes Phänomen ist. Er liefert eine theoretische Erklärung des sogenannten photoelektrischen Effektes, die ausnutzt, dass Licht Energie nur in ganz bestimmten, diskreten Mengen abgeben kann.
- **1913:** Bohrsches Atommodell.
  Atome sind im Rahmen des klassischen Modells eigentlich instabil, beobachtet wird aber Stabilität.

Bohr löst dieses Problem mit Hilfe der Annahme, dass auch Atome **Energie nur in diskreten Mengen** aufnehmen und abgeben können.

- **192X:** Klassische Quantenmechanik.
  Theoretische **Begründung der Quantenmechanik** durch Arbeiten von Bohr, Born, de Broglie, Dirac, Fermi, Heisenberg, Jordan, Schrödinger und anderen.

- **1936:** Turing Maschine.
  Turing entwirft eine später nach ihm benannte Maschine als Modell eines rein mechanisch arbeitenden Rechengerätes. Viele ähnliche Modelle wurden danach untersucht, die sich alle als äquivalent herausstellten. Dies führte zur Church-Turing-These, nach der die im intuitiven Sinne berechenbaren Funktionen genau die von einer **Turing-Maschine** berechenbaren sind.

- **197X**: Probabilistische Turing-Maschine.
  Eine **probabilistische Turing-Maschine** hat zusätzlich zu ihrer normalen Eingabe Zugriff auf eine Folge von Zufallszahlen. Diese Verallgemeinerung führte zur folgenden, modernen Version der erweiterten Church-Turing-These: Als im intuitiven Sinne effizient berechenbar werden nun alle Funktionen angesehen, die von einer probabilistischen Turing-Maschine mit beschränktem Fehler in Polynomialzeit berechenbar sind. Generell lassen sich deterministische und probabilistische Turing-Maschinen durch einfache Rechenmaschinen realisieren, die nach den Gesetzen der **klassischen Physik** arbeiten. Wie sieht es aber aus, wenn man die klassische Physik verlässt und zur **Quantenphysik** übergeht?

- **1982:** Feynmans Vermutung.
  Der Physiker und Nobelpreisträger Richard Feynman weist darauf hin, dass klassische Rechner vermutlich inhärente Schwierigkeiten haben, effizient beliebige physikalische Systeme zu simulieren, so wie sie durch die Quantenmechanik beschrieben werden. Konkret gibt er dann selbst die Antwort auf die Frage, ob klassische Computer Quantensysteme simulieren können: Er zeigt, dass ein Quantensystem durch eine probabilistische Turing-Maschine lediglich bei einer exponentiellen Abnahme der Geschwindigkeit simuliert werden kann. Anders formuliert: **Quantencomputer könnten exponentiell schneller sein als klassische Computer!**

- **1985:** Deutschs quantenmechanisches Rechenmodell.
  Der israelisch-britische Physiker David Deutsch entwickelt ein formales Modell für einen Rechner, der in der Lage sein soll, beliebige physikalische Systeme effizient zu simulieren. Sein Modell ist das quantenmechanische Analogon zur klassischen Turing-Maschine, die **Quanten-Turing-Maschine (QTM)**. Da das Modell formal etwas unhandlich ist, werden ersatzweise häufig sogenannte Quantenschaltkreise verwendet. Man kann beweisen, dass geeignete Varianten von Quantenschaltkreisen effizient durch Quanten-Turing-Maschinen simuliert werden können und umgekehrt.

- **1994:** Shor Algorithmus.
  Der amerikanische Mathematiker und Informatiker Peter Shor entwirft einen Algorithmus für QTMs, der in polynomialer Zeit ganze Zahlen faktorisieren kann. Von diesem Problem wird vermutet, dass es von klassischen Rechnern nicht in Polynomialzeit gelöst werden kann (allerdings kann man das bis heute nicht beweisen). Der **Algorithmus von Shor** hat wichtige Auswirkungen insbesondere für die Kryptografie, da die Sicherheit moderner asymmetrischer Kryptografie-Verfahren wie RSA auf der Annahme beruht, dass das Faktorisierungsproblem nicht effizient lösbar ist.

- **1996:** Grover Algorithmus.
  Der indisch-amerikanische Informatiker Lov Grover stellt einen Algorithmus für die Suche in einer Datenbank mit ungeordneten Einträgen vor, der eine quadratische Beschleunigung im Vergleich zu

den bekannten klassischen Algorithmen liefert. Es hat sich später herausgestellt, dass sowohl Shors Algorithmus als auch der **Algorithmus von Grover** prinzipiell auf große Klassen ähnlicher Probleme übertragbar sind.

- **1998:** Erste experimentelle Quantencomputer.
  Vorstellung eines funktionierenden 2**-Qubit-Quantencomputers** durch Jonathan Jones und Michele Mosca an der Oxford University und kurz danach durch Isaac Chuang am IBM Almaden Research Center unter Mitwirkung der Stanford University und des MIT.

- **1999:** Gründung von D-Wave Systems.
  Mit **D-Wave Systems** betritt erstmals ein Unternehmen den Markt, welches sich primär dem Quantum Computing sowie der Entwicklung und dem Vertrieb von Quantum Computing Soft- und Hardware verschrieben hat.

- **2001:** Faktorisierung von 15 durch Shor Algorithmus.
  Erste Anwendung des Shor Algorithmus zur **Faktorisierung von 15** am IBM Almaden Research Center und der Stanford University unter Einsatz von speziellen Molekülen, mit deren Hilfe durch Spins die erforderlichen Qubits realisiert werden.

- **2005:** Erstes Qubyte.
  Die experimentelle Realisierung des ersten kompletten **Qubytes** wird berichtet vom Institut für Quantenoptik und Quanteninformation an der Universität Innsbruck.

- **2009:** Quantenverschränkung über ersten größeren Abstand.
  Physiker am National Institute of Standards and Technology (kurz NIST) können erstmals **Verschränkungen** über einen makroskopischen und nicht mehr rein atomaren Abstand realisieren und so die Hoffnung auf einen auch praktisch unter nicht zu exotischen Bedingungen realisierbaren Quantencomputer wecken.

- **2011:** Erster kommerzieller Quantencomputer von D-Wave Systems.
  Die kanadische Firma D-Wave Systems gibt die Verfügbarkeit des ersten kommerziellen Quantencomputers **D-Wave One** bekannt. Er arbeitet mit 128 Qubits und kann spezielle Probleme lösen, ist aber kein universell programmierbarer Quantencomputer und unter Experten ist umstritten, ob für ihn der Name Quantencomputer überhaupt angemessen ist.

- **2012:** Quantenteleportation über ersten größeren Abstand und Gründung von 1QBit.
  Im September 2012 veröffentlicht das Wissenschaftsmagazin Nature einen Bericht über eine **Quantenteleportation** über eine Entfernung von 143 km von La Palma nach Teneriffa.
  The world's first dedicated quantum computing software company named **1QB Information Technologies (1QBit)** *is founded. Working in partnership with D-Wave Systems, Inc., 1QBit is focused on solving challenging computational problems in finance by leveraging and expanding the abilities of groundbreaking hardware that harnesses the unique properties of quantum mechanics.*

- **2013:** Gründung des QuAIL durch Google.
  Auf Initiative von Google wird das **Quantum Artificial Intelligence Lab** (auch **Quantum AI Lab** oder schlicht **QuAIL** genannt) gegründet mit dem Ziel, die Potentiale des Quantum Computings in Hinblick auf Maschine Learning und weitere komplexe AI-Probleme zu erforschen.

- **2016:** NIST-Aufruf für neue Public-Key-Krypto-Verfahren und Microsofts $LIQUi|>$-Framework.
  The National Institute of Standards and Technology (NIST) is now accepting submissions for **quantum-resistant public-key cryptographic algorithms**. *The deadline for submission is November 30, 2017. Please see the Post-Quantum Cryptography Standardization menu … for the complete submission*

requirements and evaluation criteria.
LIQUi|> is a simulation platform to aid in the exploration of quantum computation. LIQUi|> stands for **Language-Integrated Quantum Operations**. *A quantum operation is usually referred to as a unitary operator U applied to a column state vector (also known as a ket-vector). The i is just a constant scaling factor, hence the acronym. LIQUi|> includes a programming language, optimization and scheduling algorithms, and quantum simulators. It can be used to translate a quantum algorithm written in the form of a high-level program into the low-level machine instructions for a quantum device. LIQUi|> is being developed by the Quantum Architectures and Computation Group (QuArC) at Microsoft Research.*

- **2017:** D-Wave 2000Q, IBM-Q-Projekt, EU Quantum Technologies Flagship und Chinas Aktivitäten.
  D-Wave Systems gibt die kommerzielle Verfügbarkeit des **D-Wave 2000Q Quantum Annealer** bekannt, der mit 2048 Qubits arbeitet und spezielle Probleme lösen kann, aber wohl kein universell programmierbarer Quantencomputer ist.
  IBM gibt im Rahmen seines **IBM-Q-Projekts** die erfolgreiche Konstruktion und den Test von zwei universellen Quantencomputer-Prozessoren bekannt: Eine 16 Qubit-Architektur für den allgemeinen Einsatz (online zum Testen verfügbar) und eine erweiterte 17 Qubit-Architektur für den kommerziellen Gebrauch.
  Im Rahmen des neuen EU-Flaggschiffs zu Quantentechnologien kündigt die EU-Kommission die erste Ausschreibung an. Das **Quantum Technologies Flagship** ist das dritte Flaggschiffprojekt der EU im Bereich Forschung, nach den ersten beiden Flaggschiffen Human Brain Project und Graphene. Das Flaggschiffprojekt wird ein Budget von einer Milliarde Euro haben und soll neuartige Entwicklungen im Computing- und Kommunikationsbereich unterstützen, die auf Quantentechnologien basieren.
  Auf einem Areal von rund 370000 m^2 Größe in Hefei, Anhui Province, China, entsteht für geschätzt 10 Milliarden Dollar das chinesische **National Laboratory for Quantum Information Sciences**. Es soll planmäßig im Jahre 2020 fertiggestellt sein und sich zwei zentralen Aufgaben widmen: Quantum Metrology und Quantum Computing.

- **2018:** Tangle-Lake- und Bristlecone-Chips sowie National Quantum Initiative Act.
  Intel berichtet über die erfolgreiche Konstruktion eines 49-Qubit-Chips mit Namen **Tangle Lake** und Google gibt die Entwicklung eines 72-Qubit-Chips mit Namen **Bristlecone** bekannt. Ziel beider Projekte ist es, mit Hilfe dieser Chips erstmals konkret Anwendungsprobleme angehen zu können, die selbst für existierende klassische Super-Computer nicht lösbar sind.
  Die amerikanische Regierung verabschiedet den **National Quantum Initiative Act**, in dessen Rahmen allein für die kommenden fünf Jahre etwas mehr als eine Milliarde Dollar für Forschung und Entwicklung im Bereich der Quantentechnologien zur Verfügung gestellt werden.

- **2019:** QuNET-Initiative des Bundesministeriums für Bildung und Forschung.
  Das deutsche Bundesministerium für Bildung und Forschung kündigt die Initiative **QuNET** an, die ein hochsicheres Netz auf Grundlage der Quantenkommunikation für die Bundesregierung entwickeln soll und im Herbst des Jahres 2019 starten wird. Dafür stellt die Regierung in der laufenden Legislaturperiode 650 Millionen Euro bereit. Die Initiative QuNET soll gemeinsam mit weiteren Maßnahmen den Grundstein für eine Quantenindustrie und Quanten-IT in Deutschland legen.

2 Mathematische Grundlagen

2.1 Einführung

Im Folgenden stellen wir in aller Kürze die wesentlichen mathematischen Grundlagen bereit, die zum Verständnis des Quantum Computings erforderlich sind. Für die Details sei auf entsprechende Lehrbücher zur linearen Algebra verwiesen, also z.B. [16], [20] oder [21].

2.2 Zahlenmengen

Definition 2.2.1 Natürliche, ganze und rationale Zahlen
Die **natürlichen, ganzen und rationalen Zahlen** *sind der Reihe nach wie folgt definiert:*

$$\begin{aligned} \mathbb{N} &:= \{0,1,2,3,4,\ldots\}\,, \\ \mathbb{Z} &:= \{\ldots,-2,-1,0,1,2,\ldots\}\,, \\ \mathbb{Q} &:= \left\{\tfrac{p}{q} \mid p,q \in \mathbb{Z}, q \neq 0\right\}\,. \end{aligned}$$

Die entsprechenden Zahlenmengen ohne die Null werden durch die Ergänzung eines Sterns definiert, also

$$\mathbb{N}^* := \mathbb{N}\setminus\{0\}\,,\ \ \mathbb{Z}^* := \mathbb{Z}\setminus\{0\}\,,\ \ \mathbb{Q}^* := \mathbb{Q}\setminus\{0\}\,.$$

Hinzu kommen natürlich noch die **reellen Zahlen** $\mathbb{R}$, die durch Vervollständigung der rationalen Zahlen entstehen (alle konvergenten Cauchy-Folgen in $\mathbb{Q}$ bekommen so einen Grenzwert zugeordnet). Auch bei den reellen Zahlen nutzen wir die obige Notation zum Ausschluss der Null, also

$$\mathbb{R} := \text{“ reelle Zahlen ”} \quad \text{und} \quad \mathbb{R}^* := \mathbb{R}\setminus\{0\}\,.$$

2.3 Komplexe Zahlen

Definition 2.3.1 Komplexe Zahlen
Es sei $\vec{a} \in \mathbb{R}^2 \setminus \{\vec{0}\}$ *beliebig gegeben. Ferner sei* $r := |\vec{a}|$ *der* **Betrag** *von* $\vec{a}$ *(auch* **Modul** *oder* **Länge** *genannt) und* $\varphi := \sphericalangle(\vec{a},\vec{e}^{(1)})$ *der gegen den Uhrzeigersinn gemessene* **Winkel** *zwischen* $\vec{a}$ *und* $\vec{e}^{(1)} = (1,0)^T$ *(auch* **Phase** *oder* **Argument** *genannt). Dann gilt*

$$\vec{a} = (a_1,a_2)^T = a_1 + \mathrm{i}a_2 = r\left(\cos(\varphi) + \mathrm{i}\sin(\varphi)\right) = re^{\mathrm{i}\varphi}\,,$$

und die unterschiedlichen Darstellungen heißen der Reihe nach **kartesische Darstellung**, **komplexe Darstellung**, **trigonometrische Darstellung** *und* **exponentielle Darstellung** *von* $\vec{a}$. *Ferner bezeichnet man* a_1 *als den* **Realteil** *von* $\vec{a}$ *und* a_2 *als den* **Imaginärteil** *von* $\vec{a}$, *kurz* $\mathrm{Re}(\vec{a}) := a_1$ *und* $\mathrm{Im}(\vec{a}) := a_2$. *Ist* $\mathrm{Re}(\vec{a}) = 0$, *dann heißt* $\vec{a}$ **rein imaginär**, *ist* $\mathrm{Im}(\vec{a}) = 0$, *dann heißt* $\vec{a}$ **rein reell**. *Schließlich schreibt man, wenn der komplexe Aspekt im Vordergrund steht, für den Vektor* $\vec{a}$ *die* **komplexe Zahl** z *und bezeichnet* $\mathbb{R}^2$ *als die* **(Menge der) komplexen Zahlen** *(oder als die* **komplexe Zahlenebene** *oder auch als die*

Gaußsche Zahlenebene), *kurz* $\mathbb{C}$, *also*

$$\boxed{\mathbb{C} := \{z = a_1 + \mathrm{i}a_2 \mid a_1, a_2 \in \mathbb{R}\}\ .}$$

Ist schließlich $z = a_1 + \mathrm{i}a_2$ *eine beliebige komplexe Zahl, so wird mit*

$$\boxed{\overline{z} := a_1 - \mathrm{i}a_2}$$

die zu z **konjugiert komplexe Zahl** *bezeichnet. Ist* z *rein reell, dann gilt* $\overline{z} = z$*; ist* z *rein imaginär, dann gilt* $\overline{z} = -z$*. Unter Zugriff auf die konjugiert komplexe Zahl lässt sich u.a. der Betrag von* z *in kompakter Form berechnen als Wurzel des Produkts von* z *mit* $\overline{z}$*, kurz*

$$\boxed{|z| \ = \ \sqrt{\overline{z}z}.}$$

2.4 Vektoren und Matrizen

Definition 2.4.1 Vektorraum $\mathbb{C}^n$

Die Menge der geordneten n-Tupel

$$\boxed{\mathbb{C}^n := \underbrace{\mathbb{C} \times \mathbb{C} \times \cdots \times \mathbb{C}}_{n\ mal} := \{(a_1, \ldots, a_n)^T \mid a_1, \ldots, a_n \in \mathbb{C}\}}$$

und ihre Verknüpfungen gemäß

$$\boxed{\begin{aligned} (a_1, \ldots, a_n)^T + (b_1, \ldots, b_n)^T &:= (a_1 + b_1, \ldots, a_n + b_n)^T\ , \\ \lambda(a_1, a_2, \ldots, a_n)^T &:= (\lambda a_1, \lambda a_2, \ldots, \lambda a_n)^T \end{aligned}}$$

bilden den **Vektorraum** $\mathbb{C}^n$ **der komplexen** n**-Vektoren über** $\mathbb{C}$ **der Dimension** n*. Man verwendet zum Aufschrieb der Vektoren die Notationen*

$$\boxed{\vec{a} := \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} := (a_1, a_2, \ldots, a_n)^T}$$

und bezeichnet $\vec{a} \in \mathbb{C}^n$ *als einen* **komplexen** n**-Vektor**.

Definition 2.4.2 Vektorraum $\mathbb{C}^{m \times n}$

Die Menge der als n Spalten hintereinander geschriebenen komplexen m-Vektoren

$$\boxed{\mathbb{C}^{m \times n} := \underbrace{\mathbb{C}^m \times \mathbb{C}^m \times \cdots \times \mathbb{C}^m}_{n\ mal} := \{(\vec{a}_1, \ldots, \vec{a}_n) \mid \vec{a}_1, \ldots, \vec{a}_n \in \mathbb{C}^m\}}$$

und ihre Verknüpfungen gemäß

$$\boxed{\begin{aligned} (\vec{a}_1, \ldots, \vec{a}_n) + (\vec{b}_1, \ldots, \vec{b}_n) &:= (\vec{a}_1 + \vec{b}_1, \ldots, \vec{a}_n + \vec{b}_n)\ , \\ \lambda(\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_n) &:= (\lambda\vec{a}_1, \lambda\vec{a}_2, \ldots, \lambda\vec{a}_n)\ , \end{aligned}}$$

bilden den **Vektorraum** $\mathbb{C}^{m \times n}$ **der komplexen** (m,n)**-Matrizen über** $\mathbb{C}$ **der Dimension** $(m+n)$*. Man*