Quantensimulatoren-Katalog

Übersicht von wichtigen Quantensimulatoren

Arbeitspapier des DFN WiN-Labors in Erlangen, Stand 15.07.2022

Inhaltsverzeichnis

Überblick

Der Begriff Quantensimulator ist nicht eindeutig und wird für analoge Quantensimulatoren, die auf Basis von Quantenphänomenen arbeiten, wie auch für Software zur Simulation von Quantencomputern bzw. -Effekten benutzt. Daher ist es zunächst nötig, die korrekten Fachbegriffe und deren Bedeutung darzustellen.

Analoge Quantensimulatoren

Analoge Quantensimulatoren sind Systeme, um das Verhalten eines Quantensystems und seiner Quanteneffekte durch ein anderes, besser kontrollierbares System zu simulieren.

Eingesetzt werden analoge Quantensimulatoren u.a. zur Analyse des Verhaltens von Molekülen, was zu einer schnelleren Medikamentenentwicklung führen kann. In der Materialforschung liegt das Potential von Quantensimulatoren in der Erforschung neuer Katalysatoren z.B. für das Haber-Bosch Verfahren zur Erzeugung von Ammoniak: Mit besseren Katalysatoren könnten die Energiekosten für dieses Verfahren deutlich reduziert werden.

Analoge Simulatoren haben den Vorteil, dass sie den gleichen physikalischen Systemen und Gesetzen unterworfen sind wie das zu analysierende Zielsystem. Daher arbeiten sie im Vergleich zu (digitalen) Supercomputern effektiver und können Probleme schneller lösen.

Eine Besonderheit stellen die sog. Quantenannealer der Firma D-Wave aus Kanada dar. Diese nutzen quantenmechanische Effekte aus, um Lösungen für Optimierungsaufgaben zu erhalten, für die ein klassischer Rechner zu viel Zeit benötigten würde. Der Vorteil von Quantenannealern ist im Gegensatz zu analogen Quantensimulatoren die freie Einstellbarkeit. Jedoch lassen sich diese nicht universell einsetzen, sondern sind nur für Optimierungsaufgaben geeignet.

Ein Beispiel für die Entwicklung eines analogen Quantensimulators ist das Ziel des Projekts PASQUanS im Rahmen des EU Quantum Flagship Projekts. Im Rahmen von PASQUanS soll ein analoger Quantensimulator mit 1000 Atomen oder Ionen entstehen, welcher universell programmierbar sein soll.

Quantencomputer-Emulator

Da es bisher noch keinen Quantencomputer gibt, der sich universell für beliebige Arten von Problemen programmieren lässt und der Einsatz solcher Computer zur Lösung praktischer Aufgaben erst am Anfang steht, werden sog. Quantencomputer-Emulatoren eingesetzt. Diese simulieren das Verhalten eines Quantencomputers und die dazugehörigen QuBits auf einem klassischen Computer und erlauben es, Softwarelösungen und Algorithmen für (zukünftige) Quantencomputer bereits jetzt zu testen und zu entwickeln. Im Folgenden beziehen sich die

Begriffe Quantensimulator bzw. –Simulation ausschließlich auf Quantencomputer-Emulatoren; reine analoge Simulatoren werden in diesem Dokument nicht weiter betrachtet.

Quantensimulatoren werden z.B. eingesetzt, um das Verhalten und den Einfluss von Quantentechnologien auf Bereiche wie Kommunikationsnetze, Cyber-Sicherheit und Rechenleistung zu untersuchen. Da die Einbindung von Anwendungen und Konzepten auf Basis von Quantenhardware in konventionelle Technologien noch am Anfang steht, bieten Quantensimulatoren bereits jetzt die Möglichkeit, Vorhersagen und Auswirkungen dieser zukünftigen Technologie zu analysieren.

Mittlerweile gibt es eine Vielzahl von Anbietern und Software für Quantensimulatoren. Vor allem große Konzerne (IBM, Google, Amazon, Microsoft) die selbst real existierende Quantencomputer herstellen und weiterentwickeln, bieten –überwiegend kostenlos– Quantensimulatoren mit umfassenden Bibliotheken und grafischen Tools an. Nach vorheriger kostenloser Registrierung können beispielsweise mit dem von IBM entwickelten Simulator Qiskit Programme auf einem Quantencomputer von IBM ausgeführt werden.

Generell ergeben sich beim Arbeiten mit QuBits (nicht simuliert) einige Besonderheiten: Dazu zählt z.B., dass Quantencomputer anfällig für Fehler sind, welche u.a. aus thermischem Rauschen, Verlust von Photonen etc. resultieren. Die Folge ist, dass die Zustände von QuBits manipuliert werden und folglich Fehler bei den Berechnungen entstehen. Es gibt zwar auch Softwarepakete zur Simulation von Quantenfehlern- dennoch kann eine Simulationssoftware das reale Hardwaresystem- aufgrund äußerer Einflüsse nur bedingt abbilden.

Je nach Einsatzgebiet lassen sich Quantensimulatoren in verschiedene Kategorien einteilen:

Quantenschaltungssimulatoren

Diese Art von Simulatoren eignet sich dafür, grundlegende Eigenschaften von Quantentechnologien wie einzelne QuBits, Verschränkung oder Quantenteleportation zu simulieren.

Quantennetzwerksimulatoren

Diese Simulatoren sind dafür gedacht, vereinfacht Netzwerke zu simulieren, die mit Quantenhardware-Komponenten betrieben werden (sollen) und sind z.T. auch in der Lage, die Übertragung von Quantenzuständen bis zur physikalischen Schicht zu modellieren.

QKD-Simulatoren

QKD (Quantum Key Distribution) Simulatoren dienen auf der Anwendungsschicht zur Nachbildung der Schlüsselerzeugung, der Übertragung und dem Austausch von Schlüsseln. Da es sich bei einem QKD-Schlüssel um eine Zufallszahl handelt, eignet sich dieses Simulationsverfahren auch als Zufallszahlengenerator.

Kapitel A: Simulation mit Plattformen

In diesem Kapitel sind Online-Plattformen ausgeführt, die es ermöglichen, Simulationen online zu erstellen und ggf. vom privaten PC aus hochzuladen. Darüber hinaus bieten viele Plattformbetreiber nach Registrierung einen Zugriff auf reale Quantenhardware z.B. Quantenprozessoren und „klassische“ Simulatoren für mehrere QuBits an. Je nach Anbieter fallen für die Nutzung dieses Service auch Kosten an.

Amazon-Braket

Plattform	Amazon-Braket
Eigenschaften	• Amazon Braket ist eine Plattform von AWS (Amazon Web-Services) der das Arbeiten mit verschiedenen Arten von Quantencomputern und Schaltungssimulatoren ermöglicht • Erstellen von Quantenprojekten in einer Cloud • Ausführen auf realer Quantenhardware möglich • Bietet Beratung und Support für Benutzer bzw. Unternehmen über das Quantum Solutions Lab an • Ermöglicht die Benutzung von Quantencomputern bzw. Prozessoren von D:Wave, IonQ, Rigetti, OQC und in Kürze auch von QuEra • Für Wissenschaftler bietet AWS ein Cloud Credit Programm an
Einsatzgebiete	• Zum Entwickeln und Bauen von Quantenalgorithmen • Testen von Algorithmen mit Quantensimulatoren • Verknüpfung mit anderen AWS-Produkten • AWS-Braket Services werden auch von Unternehmen verwendet, um Quantentechnologien bzw. Produkte für Kunden zu entwickeln (s. hier)
Sprache	• Das Braket SDK (Software Development Kit) ist eine in Python geschriebene Bibliothek • Braket SDK ist OpenSource
Lizenz	• Amazon Braket SDK: Apache 2.0 Lizenz
Zugang	• Installation und Download des Braket SDK ist kostenlos • Nutzung von Simulatoren und Quantenhardware i.d.R kostenpflichtig
OS	• Plattformunabhängig: Python Interpreter erforderlich
Repository	• GitHub Repository des Amazon-Braket SDK
Dokumente	• Beispiele zu Braket

Azure Quantum (Q#/QKD)

Plattform	Azure-Quantum
Eigenschaften	• Plattform des Microsoft Cloud Dienstes Azure, mit der Nutzer Quantensoftware erstellen können • Bietet Zugang zu Rechnern auf Basis von Ionenfallen der Hersteller IonQ und Honeywell • Stellt Optimierungsalgorithmen für Quantenannealing (ohne Hardware) bereit
Einsatzgebiete	• Simulation von Quantenschaltungen, Optimierungsaufgaben, Simulation eines vollständigen Quantencomputers mit dem Fullstack Simulator (QKD)
Sprache	• Q# • Python
Lizenz	
Zugang	• Azure Konto kann hier kostenlos erstellt werden • Zugriff auf Quantenhardware bzw. Buchen von Rechnerressourcen kostenpflichtig
OS	• Azure Pakete lassen sich plattformunabhängig installieren
Repository	• Bibliotheken von Q#
Dokumentation	• Tutorials und Dokumentationen zu Q#

Google Quantum AI (Cirq)

Plattform	Google Quantum AI
Eigenschaften	• Google Quantum AI (GQA) enthält eine Sammlung von Tools für die Entwicklung von Quantenalgorithmen • Quantenalgorithmen für Simulatoren und Quantenhardware werden mit der Python Bibliothek Cirq geschrieben • Es besteht die Möglichkeit, über den Quantum Computing Service in Cirq geschriebenen Programme auf realer Quantenhardware zu testen. • GQA enthält zudem auch zahlreiche Links zu aktuellen Forschungsgebieten und Veröffentlichungen zum Thema Quantentechnik und Bildungsangebote • Neben Cirq bietet GQA auch Bibliotheken u.a. zum Simulieren von fermionischen Systemen (OpenFermion) und eine weitere für hybrides (klassisch/Quanten) Machine Learning (TensorFlow Quantum) an
Einsatzgebiete	• Erstellung und Simulation von Quantenalgorithmen bzw. Quantenschaltungen
Sprache	• Python
Lizenz	• Cirq: Apache 2.0 Lizenz
Zugang	• Download von Cirq kostenlos • Der Quantum Computing Service steht für die Öffentlichkeit derzeit nicht zur Verfügung • Weitere Bibliotheken wie OpenFermion und TensorFlow Quantum sind ebenfalls kostenlos erhältlich
OS	• Cirq kann plattformunabhängig genutzt werden, sofern Python unterstützt wird
Repository	• Cirq auf GitHub
Dokumentation	• Dokumentation und Tutorials zu Cirq • Weitere Software bzw. Pakete von GQA: s. hier

IBM Quantum (Qiskit)

Plattform	IBM Quantum
Eigenschaften	• IBM Quantum ist eine Plattform von IBM für Quantentechnologien • Mit dem Quantum Composer können Quantenschaltungen online erstellt und getestet werden • Der Service IBM Quantum Lab ermöglicht es Benutzern mit Qiskit erstellte Programme online (ohne Installation) in Jupyter Notebook zu erstellen und zu testen • Auf der Webseite befindet sich auch eine Übersicht zu von IBM betriebenen Quantensimulatoren und (hybriden) –Hardwaresystemen • Enthält auch Programme zur Unterstützung von Wissenschaftlern und Lehrenden in diesem Bereich • Bietet Zugang zu realer Quantenhardware, auf der z.B. Quantenschaltungen getestet werden können
Einsatzgebiete	• Online Modellierung und Simulation von Schaltungen • Ausführung der Programme auf realer von IBM entwickelter Quantenhardware • Simulation von (existierender) Quantenhardware
Sprache	• Python Bibliothek
Lizenz	• Apache 2.0 Lizenz
Zugang	• Schaltungen können nach Registrierung auf realer Hardware getestet werden • Keine Installation bei Online Nutzung erforderlich, jedoch ist Registrierung erforderlich • Registrierung ist kostenlos
OS	• Plattformunabhängig: für Zugriff auf Plattform lediglich Browser erforderlich
Repository	• Übersicht Qiskit Repositories
Dokumentation	• Seite von Qiskit • Dokumente zu Projekten von IBM Quantum

QUANTASTICA (QPS /Qubit Toaster)

Plattform	QUANTASTICA
Eigenschaften	• Organisation, die unterschiedliche Softwaretools und Lösungen im Bereich Quantencomputing entwickelt und bereitstellt • Seit Mai 2019 ist QUANTASTICA Partner von Rigetti im Bereich Anwendungsentwicklung • Gute Vernetzung mit weiteren Partnern: Unitary Zero Space, icebreaker und Unitary Fund • Zu den von QUANTASTICA entwickelten Simulatoren gehören der Qubit Toaster und quantum circuit • Mit dem Quantum Programming Studio lassen sich Online grafisch Simulationen durchführen und Quantenalgorithmen erstellen • Der QConverter -als Online oder Commandline Version- ist ein Tool, mit dem sich Quanten- Programmiersprachen für spezifische Quantencomputer in andere umwandeln lassen z.B.QASM nach PyQuil • Quantum Algorithm Generator: Mit diesem Tool werden Quantenschaltungen anhand von Statusvektoren erzeugt (Reverse Engineering). Installationsanleitung befindet sich hier.
Einsatzgebiete	• Je nach verwendetem Softwarepaket lassen sich unterschiedliche Aufgaben wie Codekonvertierung, Erstellen von Simulationen bzw. Algorithmen erfüllen oder Quantenschaltungen aus Statusvektoren ableiten
Sprache	• Quantum Circuit Simulator, QConvert: Javascript • Quantum Algorithm Generator: Python • Qubit Toaster: k.A.
Lizenz	• Quantum Algorithm Generator: Apache 2.0 Lizenz • QConverter, Quantum Circuit: MIT Lizenz • Qubit Toaster: k.A.
Zugang	• Keine Beschränkungen außer Quantum Algorithm: Login bei Quantum Programming Studio erforderlich
OS	• Plattformunabhängig: Python Interpreter bzw. Browser (Javascript) erforderlich
Repository	QConverter: GitHub Quantum Algorithm Generator, Qubit Toaster: nicht vorhanden Quantum Circuit: GitHub
Dokumentation	Qubit Toaster: s. hier Quantum Circuit: s. hier Quantum Algorithm Generator: s.hier QConverter: s. hier

Quantum Programming Studio

Simulator	Quantum Programming Studio
Eigenschaften	• Das Quantum Programming Studio (QPS) ist eine webbasierte grafische Entwicklungsumgebung für Quantenalgorithmen und Simulationen • Schaltungen können einfach via Drag & Drop erstellt werden • QPS ist eine Partner Anwendung von Rigetti • Quantenschaltungen können als Format für unterschiedliche Sprachen bzw. Frameworks exportiert und auf verschiedenen Simulatoren und Quantencomputer ausgeführt werden wie TensorFlow Quantum oder Amazon Braket • QPS ist eine Paket Erweiterung des Opensource Quantum Circuit Simulator
Einsatzgebiete	• Durchführen von Simulationen, Erstellung von Algorithmen für verschiedene Plattformen
Sprache	• Quantum Circuit Simulator: JavaScript
Lizenz	• MIT-Lizenz
Zugang	• Für die Benutzung des Webinterface ist eine kostenlose Registrierung erforderlich
OS	• OS unabhängig: Es ist lediglich ein Browser mit Javascript erforderlich
Respository	• Auf GitHub verfügbar
Dokumentation	• Dokumentation zu unterschiedlichen Themen von QPS

Quantum Inspire

Plattform	Quantum Inspire
Eigenschaften	• Quantum Inspire (QI) ist eine von QuTech entwickelte Multi-Hardware Quantentechnologie Plattform • Weitere Partner des Projekts sind hier aufgelistet • Bietet die Möglichkeit, eigene Quantenalgorithmen auf Simulatoren oder realer Quantenhardware zu testen • Algorithmen müssen in der Sprache cQASM verfasst sein, die auch grafisch (z.B. in Form von Quantenschaltungen) angezeigt werden können. • QX ist ein Quantencomputeremulator auf dem –abhängig vom Account-Status- bis zu 34 QuBits simuliert werden können (Emulator Backends) • Als Hardware-Backends stehen zwei Quantenprozessoren zur Verfügung: der Spin-2 mit 2 Qubits und der Starmon-5 mit fünf Qubits
Einsatzgebiete	• Testen von (Quanten-) Algorithmen auf realer Quantenhardware oder Quantencomputer Emulatoren
Sprache	• cQASM
Lizenz	• Benutzerrichtlinien und Nutzungsbedingungen befinden sich hier.
Zugang	• Es gibt drei Arten von Nutzeraccounts: • **Anonymus**: keine Registrierung erforderlich, Nutzung des cQASM Online Editors und Simulation von bis zu fünf QuBits auf Simulator Backend möglich. Keine Speicherung von Projekten möglich • **Basic Account**: Registrierung erforderlich. Speicherung von Projekten möglich. Zugriff auf Quantenhardware bzw. –Prozessoren und Nutzung von unterschiedlichen Emulator Backends möglich • **Advanced/Special Account**: für spezielle Anliegen wie der Registrierung eine Gruppe von Personen oder Erweiterung des bisherigen Accounts z.B. für den Zugriff auf mehr QuBits. Dies kann über das Account Management im Benutzerkonto beantragt werden
OS	• plattformunabhängig: Browser erforderlich
Repository	• kein Repository verfügbar
Dokumentation	• API Referenz, Quick Start-Guide, cQASM-Handbuch und Codebeispiele sind hier verfügbar

Quantum Network Explorer (NetSquid)

Simulator	Quantum Network Explorer
Eigenschaften	• Ermöglicht das interaktive Testen von Anwendungen in Quantennetzen wie QKD oder verteilte CNOT Operationen • Entwickelt von Mitarbeitern des niederländischen Forschungsprojekts QuTech zum Thema Quantentechnologie • Enthält auch eine Dokumentation zu verschiedenen Themengebieten rund um Quantennetze wie QKD, Sicherheit, Quantencloud etc… • Für die Entwicklung von eigenen Anwendungen für den Quantum Network Explorer kann die Software QNE-ADK heruntergeladen werden
Einsatzgebiete	• Geeignet, um sich einen schnellen Überblick zum Thema Quantennetze zu verschaffen und eigene kleine Experimente durchzuführen z.B. QKD Übertragung von Amsterdam nach Rotterdam
Sprache	In Python geschrieben- CLI Interface
Lizenz	MIT-Lizenz
Zugang	• Login erforderlich, falls Versuche mit dem QNE online gespeichert werden sollen • Es ist eine Registrierung für den Simulator NetSquid erforderlich, falls QNE lokal installiert werden soll. Grund dafür ist, dass QNE auf Bibliotheken dieses Simulators zurückgreift
OS	• Browser für Online Zugriff erforderlich • MacOS oder Linux mit Python 3.7 und pip Version 19
Respository	• QNE-ADK in GitHub
Dokumentation	• Informationen zu Quantennetzen, Quantentechnologie und Userguides zu QNE bzw. QNE-ADK sind hier abrufbar

Rigetti QCS (pyQuil)

Plattform	Rigetti QCS
Eigenschaften	• Der Rigetti Quantum Cloud Service ermöglicht den Zugriff auf QPUs (Quantenprozessoren) von Rigetti • Das Forest SDK enthält Softwaretools zum Erstellen von Programmen in Quil, die dann mittels QCS oder eines Simulators ausgeführt werden • Das Forest SDK besteht aus drei Komponenten: • pyQuil: Python Bibliothek zum Erstellen und Ausführen von Programmen in Quil • quilc: Compiler für Quil Programme • QVM: virtuelle Maschine zur Simulation eines Quantencomputers
Einsatzgebiete	• Erstellen von Simulationen bzw. Programmen in Quil • Nutzung von Rigetti's QPUs
Sprache	• pyQuil: Python • quilc, QVM: Common Lisp
Lizenz	• pyQuil, quilc : Apache 2.0 Lizenz • QVM: Rigetti Lizenz Datei
Zugang	• Für QCS ist Registrierung erforderlich • Ansonsten keine Beschränkungen
OS	• Plattformunabhängig: Python Intepreter bzw. Browser erforderlich
Repository	• pyQuil in GitHub • quilc in GitHub • QVM in GitHub
Dokumentation	• Userguides auf QCS Homepage • (API) Referenzen auf Homepage

Kapitel B: Simulatoren zur Selbstinstallation

Simulatoren zur Selbstinstallation sind meist kostenlos und ohne Registrierung frei erhältlich. Sie eigenen sich vor allem, um sich einen ersten Überblick in die Programmierung und Simulation mit QuBits zu verschaffen. Neben der Erstellung von (grafischen) Quantenschaltungen und Algorithmen bieten QKD- und Quantennetzwerksimulatoren die Möglichkeit, die Auswirkung und den Einfluss dieser neuen Technologie auf derzeitige Computernetzwerke näher zu untersuchen.

Quanten-Schaltungssimulatoren

blueqat

Simulator	blueqat
Eigenschaften	• In Python geschriebener Simulator mit grafischer Ausgabe • Gut geeignet, um Schaltungen, Qubit Zustände etc… zu illustrieren
Einsatzgebiete	• bluegat eignet sich zum Definieren von Schaltungen, Anwendungen im Bereich des Machine Learnings und Optimierungsaufgaben
Sprache	• Python / Jupyter Notebook
Lizenz	• Apache 2.0 Lizenz
Zugang	• Freier Zugang (GitHub)
OS	• Plattformunabhängig: Python erforderlich
Repository	• bluegat in GitHub
Dokumentation	• Verfügt über gut beschriebene Tutorials

Cirq

Simulator/Bibliothek	Cirq
Eigenschaften	• Mit Cirq lassen sich Quantenschaltungen aufbauen und Simulieren • Quantensimulator von Google mit umfangreicher Dokumentation • Wird in der Google Quantum AI verwendet
Einsatzgebiete	• Simulation von Quantenschaltungen • Testen von Algorithmen (auf Quantenbasis)
Sprache	• Python
Lizenz	• Apache 2.0 Lizenz
Zugang	• Code auf GitHub frei verfügbar
OS	• Plattformunabhängig: benötigt Python
Repository	• GitHub
Dokumentation	• Gute Dokumentation, Tutorials und zahlreiche Beispiele verfügbar

myQLM

Simulator	myQLM
Eigenschaften	• Von Atos entwickelter Simulator zum Gestalten und Simulieren von Programmen im Bereich Quantentechnologie • Neben myQLM bietet Atos weitere Services im Bereich Quantentechnologie wie den QLaaS (Quantum Learning as a Service), Q-Score und den QLM User Club an • Stellt Schnittstellen zu weiteren bekannten Simulatoren wie ProjectQ und Qiskit bereit • API zum Implementieren von Plugins
Einsatzgebiete	• Erstellen von Quantenschaltungen und Simulationen
Sprache	• Python, Jupyter Notebook, HTML, OpenQASM
Lizenz	• Atos myQLM EULA
Zugang	• Installation ohne Registrierung möglich
OS	• Plattformunabhängig: benötigt Python Interpreter
Repository	• Auf GitHub verfügbar
Dokumentation	• Dokumentation und Beispiele befinden sich auf der Homepage von myQLM

Ocean

Simulator	Ocean
Eigenschaften	• Ocean ist eine von D-Wave entwickelte Toolbox
Einsatzgebiete	• Wird verwendet zum Lösen schwieriger Optimierungsprobleme mit Quantencomputern. • Anwendungsbeispiele bzw. Probleme sind Map Coloring, Vertex Cover, Postprocessing mit Greedy Solver
Sprache	• Python
Lizenz	• Apache 2.0 Lizenz (Ocean SDK)
Zugang	• SDK Installation ohne Beschränkungen möglich • Für die Nutzung des Leap Quantum Cloud Service ist eine Registrierung erforderlich
OS	• Plattformunabhängig: Python Interpreter erforderlich
Repository	• Bei GitHub verfügbar
Dokumentation	• Dokumente zu unterschiedlichen D-Wave Toolboxen und weiterführende Informationen sind hier verfügbar

ProjectQ

Simulator	ProjectQ
Eigenschaften	• Projekt des ETH Zürich • „High Level“ Sprache für Quantenprogramme • Modular aufgebauter und anpassbarer Compiler • Backend-Interface kann ausgewählt werden: z.B. klassische oder Q-Hardware • Besitzt auch eine „Fermilib“ Bibliothek zum Lösen von Fermionen-Problemen auf Quantencomputer • Es gibt im Backend auch die Option, das Programm für IBM-Quantum Hardware übersetzen zu lassen
Einsatzgebiete	• Für die Erstellung Q-Anwendungen geeignet, die auf mehreren Backends (z.B. klassischer Computer, Q-Hardware) laufen soll
Sprache	• Python
Lizenz	• OpenSource: Apache 2 Lizenz
Zugang	• Keine Beschränkungen:
OS	• Plattformunabhängig: Python Interpreter erforderlich
Repository	• Auf GitHub verfügbar
Dokumentation	• Dokumentationsseite von ProjectQ

pyQuil

Simulator	pyQuil
Eigenschaften	• Ist Teil des Rigetti Forest SDK • Ermöglicht das Erstellen und Ausführen von Quil Programmen in Python • Für Quil ist die Installation des Quil Compilers und der QVM (Quantum Virtual Machine) erforderlich
Einsatzgebiete	• Das Forest SDK bzw. pyQuil wird im Rahmen des QCS (Quantum Cloud Service) dazu verwendet, Nutzern die Möglichkeit zur Verwendung von QPU’s (Quantenprozessoren) via QCS zu bieten
Sprache	• Python
Lizenz	• pyQuil: Apache 2.0 Lizenz
Zugang	• Installation von pyQuil ohne Registrierung möglich • Für die Benutzung von QCS ist eine Registrierung erforderlich
OS	• pyQuil: plattformunabhängig. Python Interpreter erforderlich
Repository	• pyQuil: Auf GitHub verfügbar
Dokumentation	• verschiedene Themen und Tutorials in QCS sind hier abrufbar • Dokumentation und Beispiele zu pyQuil befinden sich auf der Seite von pyQuil Homepage

Microsoft QKD (Q#)

Simulator	QDK
Eigenschaften	• Das Microsoft Quantum Development Kit (QDK) enthält vier Simulatoren für Q# Programme • Q# ist eine von Microsoft entwickelte Programmiersprache mit eigener Syntax speziell für die Programmierung im Bereich Quantentechnik • Q# und Quantum Development Kit auch ohne Azure Abonnement nutzbar • Mit Q# bzw. QDK lassen sich Anwendungen für Quantum Azure erstellen • Q# Programme lassen sich auch mit Jupyter Notebook ausführen
Einsatzgebiete	• Fokus liegt auf der Entwicklung komplexer Anwendungen bzw. Algorithmen (nicht nur einfacher Schaltungen)
Sprache	• Q# besitzt eine eigene Syntax angelehnt an Python, C#, F#
Lizenz	• Microsoft (QDK)
Zugang	• Steht kostenlos zum Download bereit • Q# ist Open Source • Auch Online (ohne Download) über Visual Studio Codespaces nutzbar, allerdings ist dieser Service nicht kostenlos • QDK lässt sich über Visual Studio und Visual Studio Code nutzen
OS	• Plattformunabhängig bei Nutzung über Visual Studio Code • Nur über Windows bei Nutzung über Visual Studio
Repository	• Quantum Bibliotheken von Microsoft auf GitHub
Dokumentation	• Download QDK • Dokumentation zu Q# und QDK

QCircuits

Simulator	QCircuits
Eigenschaften	• basiert auf dem Quantencircuit Modell • einfaches Interface für erleichterte Benutzung
Einsatzgebiete	• Simulation und Untersuchung des Verhaltens von Quantencomputern • Untersuchung und Aufbau des Verhaltens von Q-Schaltungen
Sprache	• Python-Paket
Lizenz	• MIT Lizenz
Zugang	• Installation ohne Registrierung
Repository	• Auf GitHub verfügbar
Dokumentation	• Tutorials, Beispiele, Dokumentationen auf der QCircuits Homepage verfübar

Qiskit

Simulator	Qiskit
Eigenschaften	• Von IBM entwickeltes Quantensimulator Framework • Entwickelte Schaltungen können auch auf realer Hardware von IBM –nach Registrierung- getestet werden (IBM Quantum) • Enthält auch ein Modul, mit dem Schaltungen grafisch dargestellt werden können • In der IBM Cloud sind weitere Schaltungssimulatoren (wie „extended Clifford, Schrödinger“) verfügbar • Noise Modelle wie Pauli, Depolarisation etc… verfügbar
Einsatzgebiete	• Modellierung und Simulation von Schaltungen • Ausführung der Programme auf realer von IBM entwickelter Quantenhardware • Simulation von (existierender) Quantenhardware
Sprache	• Hauptsächlich in Python geschriebenes Framework • weitere verwendete Sprachen: C++, OpenQASM, Typescript, Vu
Lizenz	• Open Source Framework • Apache 2.0 Lizenz
Zugang	• Download und Installation von Qiskit sind kostenlos und benötigen keine Registrierung • Registrierung bei Nutzung der IBM-Cloud und/oder Testen von Programmen auf realer Quantenhardware erforderlich
OS	• Plattformunabhängig: Python Interpreter erforderlich
Repository	• Simulatoren, weitere Tools und Projekte für Qiskit sind auf GitHub verfügbar
Dokumentation	• Qiskit Textbook mit zahlreichen Beispielen und Erklärungen

Quantum Computing Playground

Simulator	Quantum Computing Playground
Eigenschaften	• WebGL Projekt in Chrome • Befindet sich noch im experimentellen Stadium • Simuliert einen Quantencomputer (auf GPU) mit eigener Skriptsprache • Es können bis zu 22 Qbits simuliert werden • 3D Visualisierung von Q-Zuständen
Einsatzgebiete	• 3D Simulationen von Quantenzuständen und Gattern
Sprache	• Chrome Projekt: Quellcode nicht zugänglich
Lizenz	• k.A.
Zugang	• Auf google chrome ausgerichtet
OS	• plattformunabhängig: Browser erforderlich
Repository	• kein Repository verfügbar
Dokumentation	• Beispiele auf der Homepage unter Examples verfügbar

Quantum JavaScript (Q.js)

Simulator	Quantum JavaScript (Q.js)
Eigenschaften	• Simulator für Quantenschaltungen • Drag and Drop Editor für Webbrowser • Schaltungen können auch über Quellcode definiert werden, der dann als Schaltung angezeigt wird • Output der Schaltungen wird grafisch dargestellt
Einsatzgebiete	• Grafische Erstellung von Q-Schaltungen
Sprache	• Javascript, HTML, CSS
Lizenz	• MIT Lizenz
Zugang	• Über Webbrowser oder als Download via GitHub
OS	• Plattformunabhängig: Browser erforderlich
Repository	• Auf GitHub verfügbar
Dokumentation	• Beispiele und API Dokumentation auf der Q.js Homepage verfügbar

Qubit Toaster

Simulator	Qubit Toaster
Eigenschaften	• Gehört zum QUANTASTICA Projekt, welches auf die Entwicklung von Softwaretools und Lösungen im Bereich Quantencomputing anbietet • Der Qubit Toaster ist ein hochleistungsfähiger Quantenschaltungssimulator, der auf Geschwindigkeit ausgelegt ist. • Er basiert auf Algorithmen zur Schaltkreisoptimierung und effizienten Ausführung. • Er kann eigenständig oder zusammen mit gängigen Quantenprogrammier-Frameworks verwendet werden z.B. Qiskit, Quantum Programmingt Studio.
Einsatzgebiete	• Für Quantensimulationen, die mit höherer bzw. verbesserter Geschwindigkeit ausgeführt werden sollen (HPC)
Sprache	• k.A.
Lizenz	• k.A.
Zugang	• keine Beschränkungen
OS	• für Linux, MacOS und Windows verfügbar
Repository	• kein Repository: Programm wird über vorkompilierte Binärdatei installiert
Dokumentation	• Benutzung von QuBit Toaster auf Homepage beschrieben

QuEST

Simulator	QuEST
Eigenschaften	• verteilter, GPU-beschleunigter Simulator von universellen Quantenschaltungen, Zustandsvektoren und Dichtematrizen. • QuEST ist eine quelloffene und eigenständige C/C++-Bibliothek • Simulation von Dephasierung und depolarisierendes Rauschen möglich • Derselbe Code kann nahtlos auf allen Hardware-Backends eingesetzt werden, und die Simulationskosten und -genauigkeit können zur Kompilierungszeit geändert werden. • QuEST ist derzeit der einzige aktive verteilte QC-Simulator und der erste und einzige, der eine verteilte Dichtematrix unterstützt.
Einsatzgebiete	• Simulation von Quantenschaltungen, Statusvektoren und Dichtematrizen
Sprache	• C/C++, Cuda, JavaScript
Lizenz	• MIT Lizenz
Zugang	• Keine Beschränkungen: Downlaod und Installation kostenlos
OS	• Für MacOS, Linux und Windows verfügbar
Repository	• Auf GitHub verfügbar
Dokumentation	• Tutorials und Beispiele sind hier abrufbar

Qibo

Simulator	Qibo
Eigenschaften	• QIBO ist eine API für Quantensimulationen und Kontrolle von Quantenhardware
Einsatzgebiete	• Berücksichtigung der Eigenschaften von Hardwarebauteilen wie z.B. NISQs bei der Durchführung von Simulationen
Sprache	• Python
Lizenz	• Apache Lizenz Version 2.0
Zugang	• Keine Beschränkungen
OS	• Plattformunabhängig: Python Interpreter benötigt
Repository	• Auf GitHub verfügbar
Dokumentation	• Dokumentationsseite von Qibo

QuTip

Simulator	QuTIP
Eigenschaften	• QuTiP (Quantum Toolbox in Python) ist eine Open-Source-Software für die Simulation der Dynamik offener Quantensysteme. • grafische Ausgabe via Matplotlib.
Einsatzgebiete	• wird eingesetzt für effiziente numerische Simulationen von Hamilton Funktionen u.a. aus den Bereichen wie Quantenoptik, Ionenfallen
Sprache	• Python, HTML. Shell
Lizenz	• BSD-3 Clause Lizenz, keine Lizenz Gebühren
Zugang	• Keine Beschränkungen
OS	• Linux, MacOS, Windows
Repository	• Auf GitHub verfügbar
Dokumentation	• Dokumentation und Tutorials auf der Homepage verfügbar

Quirk

Simulator	Quirk
Eigenschaften	• Quirk ist ein grafischer Online-Simulator für (einfache) Quantenschaltungen. • Es existieren auch vorgefertigte Schaltungen mit der sich z.B. die Quantenteleportation simulieren lassen
Einsatzgebiete	• Eignet sich zur Demonstration bzw. Analyse des Verhaltens von Quantenschaltungen • Generierung von Quantenschaltungen per Drag & Drop
Sprache	• Quellcode bei GitHub verfügbar (JavaScript)
Lizenz	• Apache 2.0 Lizenz
Zugang	• Keine Beschränkungen (OpenSource)
OS	• Plattformunabhängig: Browser benötigt
Repository	• Auf GitHub verfügbar
Dokumentation	• Userguide und Videoanleitung verfügbar

Silq

Simulator	Silq
Eigenschaften	• Entwickelt vom ETH Zürich • Silq stellt unter den Programmiersprachen für Quantenschaltungen bzw-computer eine Hochsprache dar und ist einfacher zu handhaben als z.B. OpenQASM • Neben dem einfachen und schlichten Design, werden bereits quantenmechanischen Eigenschaften berücksichtigt um die sich der Benutzer nicht (mehr) kümmern muss, sodass der Code weniger fehleranfällig ist • Ein wichtiges Kriterium beim Quantencomputing ist die sog. Uncomputation d.h. die Rücksetzung von QuBits auf den Ausgangszustand, welche zur Speicherung von Zwischenergebnissen dienen. Diese Bits werden auch als Ancilla Bits bezeichnet. Diese Rücksetzung muss bei Quantencomputern gemacht werden, da i.d.R nur sehr begrenzt QuBits zur Verfügung stehen und diese wiederverwendet werden sollen. Bei klassichen Computern übernimmt diese Aufgabe ein Garbage Kollektor nach der Ausführung eines Programms • Intuitive Variablentypen für Quantenzustände wie Verschränkung, Überlagerungszustände etc…
Einsatzgebiete	• Programme für Quantencomputer erzeugen • Untersuchung von Quanten-Schaltungen
Sprache	• Q#, D, Tex, Python
Lizenz	FreeBSD Lizenz
Zugang	• Keine Beschränkungen: Download und Installation ohne Registrierung
OS	• Es wird VS Code benötigt um das Silq PlugIn zu installieren
Repository	• Auf GitHub verfügbar
Dokumentation	• Dokumentation und Beispiele auf der Homepage verfügbar

Strawberry Fields

Simulator	Strawberry Fields
Eigenschaften	• Von Xanadu entwickelte Python Bibliothek • Dient zum Simulieren und Ausführen von Programmen auf Quantenhardware auf Basis von Photonen • Große Auswahl an Tutorials und Beispielen aus dem Bereich Quantenphotonik • Bietet Zugang über die Xanadu Cloud zum ersten vollprogrammierbaren Photonen-Quantencomputer
Einsatzgebiete	• Programme/Simulationen für Quantenhardware basierende auf Photonen
Sprache	• Python Bibliotheken
Lizenz	• Apache 2.0 Lizenz
Zugang	• Keine Beschränkungen
OS	• Plattformunabhängig: Python Interpreter erforderlich
Repository	• Auf GitHub verfügbar
Dokumentation	• Dokumentation und Beispiele befinden sich auf der Homepage von Strawberry Fields unter Documentation

XACC

Simulator	XACC
Eigenschaften	• Framework für hybride (klassisch-quanten) Computerarchitekturen • Unterstützt Programmieren im klassischem und Quantenumfeld • Bietet die Möglichkeit zum Ausführen von Quantencode auf Quantenprozessoren von u.a. Rigetti, IBM, IonQ • Basiert auf dem C++ Micro Services Projekt
Einsatzgebiete	• Hybride Computerarchitekturen, Ausführen von Quantencode auf verschiedenen Computerarchitekturen bzw. -Prozessoren
Sprache	• C++
Lizenz	• Eclipse Public License and Eclipse Distribution License, BSD-3 Clause
Zugang	• Keine Beschränkungen
OS	• Ubuntu 16.04/18.04, Centos 7, Fedora 7/30, MacOS X • C++ Compiler und CMake erforderlich
Repository	• Auf GitHub verfügbar
Dokumentation	• Tutorials, Beispiele und weiterführende Informationen befinden sich auf der Dokumentationsseite zu XACC

Quantennetzwerksimulatoren

Interlin-q

Simulator	Interlin-q
Eigenschaften	• Simulator, der eine Master-Slave Kontrollstruktur in einem Quantennetz verwendet • Mit Interlin-q können einzelne Schaltungen als auch verteilte Algorithmen in einer Quantennetzwerktopologie definiert und simuliert werden • Quantenschaltungen werden auf eine Quantencomputer Architektur übertragen und die Kommunikation zwischen den einzelnen Nodes entsprechend dem Master-Slave Prinzip geregelt
Einsatzgebiete	• Erstellen und Testen von verteilten Quantenalgorithmen auf verschiedenen Quantencomputer Architekturen
Sprache	• Python
Lizenz	• MIT Lizenz
Zugang	• Keine Beschränkungen
OS	• Plattformunabhängig: Python Interpreter erforderlich
Repository	• Auf GitHub verfügbar
Dokumentation	• Interlin-q Dokumentationsseite

NetSquid

Simulator	NetSquid
Eigenschaften	• Modelliert den Einfluss der Zeit in Quantennetzwerken und Computersystemen • Modularer Aufbau: Einzelne Komponenten können ineinander verschachtelt werden (Quanten Computing Library)
Einsatzgebiete	• Für den Entwurf/Simulation eines quantenbasierten Internets • Für den Entwurf modular aufgebauter Quantencomputer Architekturen • Performance-Untersuchung der physikalischen Schicht (Quanten Hardware) von Quantennetzwerken
Sprache	• Python Paket
Lizenz	• MIT Lizenz
Zugang	• Kostenlos, aber Registrierung für Download erforderlich
OS	• Plattformunabhängig: Python Interpreter erforderlich
Repository	• GitHub Repository nur mit Beispielen verfügbar
Dokumentation	• Auf der Homepage von Netsquid verfügbar

OpenQL

Simulator	OpenQL
Eigenschaften	• Framework zur Quanten-Programmierung in Python/C++ • Anders als z.B. Qiskit liegt der Fokus auf der Erzeugung von Assembly Code für verschiedene (Mikro-) Architekturen von QuTech • Format des Assembly Codes ist cQASM (Quantum Assembly Language) • Format des Ausgangscodes ist plattformabhängig
Einsatzgebiete	• Zur Erzeugung von Code für QuTech Architekturen
Sprache	• C++, Python, HTML, JavaScript, OpenQASM
Lizenz	• Apache 2.0 Lizenz
Zugang	• Keine Beschränkungen
OS	• Plattformunabhängig: Python Interpreter erforderlich
Repository	• Auf GitHub verfügbar
Dokumentation	• Dokumentationsseite zu OpenQL

QuISP

Simulator	QuISP
Eigenschaften	• Event gesteuerter Simulator für Quanten-Repeater Netzwerke • Ziel des Projekts ist die Simulation von 100 Netzwerken mit je 100 Knoten • Bei derart großen Netzen ist es nicht möglich, diese auf Hamilton Operator Ebene oder CNOT Gatter zu simulieren; es werden daher nur Q-Fehlzustände erfasst und nicht der komplette Q-Status (Error-Basis) • Unterstützt alle außer „Pauli“ Fehlerarten (Unterschied zu anderen Simulatoren) • Setzen von Link-Längen im Netzwerk, Gate Fehlerraten, Speicherzustände einzelner Q-Bits • Benötigt OmNET++ und Eigen, ein Matrixrechner für C/C++ • QuISP ist ein Produkt der AQUA (Advances Quantum Architecture) Forschungsgruppe
Einsatzgebiete	• Protokoll-Design • Untersuchung des Verhaltens von großen komplexen heterogenen Netzen
Sprache	• C++, Python, Shell
Lizenz	• BSD 3 Clause Lizenz
Zugang	• Keine: Download und Installation kostenlos
OS	• Für Linux, Windows und MacOS verfügbar
Repository	• Auf Github verfügbar
Dokumentation	• Dokumentation OmNET++ • Dokumentation QuISP

QuNetSim

Simulator	QuNetSim
Eigenschaften	• Simulator für Quantennetzwerke • Stellt Framework für die Entwicklung von Protokollen in Quantennetzen bereit • Enthält bereits Quanten-Basistechnologien wie Quantenteleportation, QKD Erzeugung etc… • Benutzt die grafische networkx Bibliothek von Python zur Darstellung und Erzeugung von Netzen • Event gesteuerter Simulator • Behandelt Quantennetze wie klassische Netze in Bezug auf das Schichtenmodell
Einsatzgebiete	• Beispielsweise zum Nachverfolgen (Schritt für Schritt) einer QuBIt Teleportation im Q-Netz geeignet • Zum Ausprobieren bzw. Erstellen von Netzwerkprotokollen auf „high level“ Niveau • Nicht geeignet zur genauen Simulation von Quanteneffekten • Für Einsteiger in Quantennetze geeignet
Sprache	• Python
Lizenz	• MIT Lizenz
Zugang	• Keine Beschränkungen: Download und Installation kostenlos
OS	• Plattformunabhängig: Python Interpreter nötig
Repository	• Auf GitHub verfügbar
Dokumentation	• Dokumentation hier abrufbar

SeQUeNCe

Simulator	SeQUeNCe
Eigenschaften	• Wird verwendet, Um Effekte in Quantennetzen auf den unteren Netzwerkschichten zu analysieren wie z.B. die Zwischenspeicherung von Quantenzuständen
Einsatzgebiete	• Zum Testen von Protokollen, Netzwerkparametern und Topologien
Sprache	• C++, Python, Makefile
Lizenz	• Open Source Lizenz
Zugang	• Keine Beschränkungen
OS	• Plattformunabhängig: Python Interpreter erforderlich
Repository	• Auf GitHub verfügbar
Dokumentation	• SeQUeNCe Dokumentationsseite

SimulaQron

Simulator	SimulaQron
Eigenschaften	• Simulator für Anwendungsschicht in Quantennetzwerken • Wurde von QuTech entwickelt • Stellt Infrastruktur aus verteilten Q-Prozessoren die über Q-Kommunikationskanäle verbunden sind, bereit • Jeder Q-Prozessor ist über einen Server verfügbar, der auf einem normalen PC läuft (auch auf verschiedenen PCs verteilt); SimulaQron verbindet die Prozessoren, zur Übertragung von Q-Bits und Verschränkung über Distanzen • Auf die simulierte Hardware kann dann über Python, C-Bibliotheken oder das CQC Interface zugegriffen werden
Einsatzgebiete	• Erstellung eigener Anwendungen für das „Quanteninternet“ • Entwicklung von Software-Engineering Konzepten und Bibliotheken für Quantum-Internet
Sprache	• Python
Lizenz	• Siehe Lizenzdatei
Zugang	• Keine Beschränkungen
OS	• Plattformunabhängig: Python Interpreter erforderlich
Repository	• Auf GitHub verfügbar
Dokumentation	• Installationsanleitung • SimulaQron Dokumentation

SQUANCH

Simulator	SQUANCH
Eigenschaften	• entwickelt für die Simulation von Quantennetzwerken • wurde entwickelt im Rahmen des INQNET Programm • enthält klassische und Quanten-Fehlermodelle • Protokolle wie Quantum Error Korrektur in Beispielen enthalten (als Source Code)
Einsatzgebiete	• Zum Testen von Netzwerkprotokollen und Quantenübertragung • Simulation von Multiparty Netzwerken
Sprache	• Python
Lizenz	• MIT Lizenz
Zugang	• Keine Beschränkungen
OS	• Plattformunabhängig: Python Interpreter erforderlich
Repository	• Auf GitHub verfügbar
Dokumentation	• SQUANCH Dokumentationseite

QKD Simulatoren

QKDNetSim

Simulator	QKDNetSim
Eigenschaften	• Es handelt sich um ein QKD-Simulationsmodul, welches in den etablierten NS-3 Simulator für Netzwerke eingebettet wird • Daher kein vollständiger Quantensimulator • Es können Netzwerke auf QKD-Basis im Overlay oder TCP/IP Modus untersucht werden • Basiert auf dem NS-3 Network Simulator (s. hier) • QKDNetSim kann auch über ein Webinterface genutzt werden
Einsatzgebiete	• Geeignet, um die Auswirkungen bzw. Verhalten in herkömmlichen Netzen zu untersuchen
Sprache	• C/C++, Python, Perl
Lizenz	• GPL 2.0 Lizenz
Zugang	• Keine Beschränkungen
OS	• Linux
Repository	• Auf GitHub verfügbar
Dokumentation	• QKDNetSim Dokumentation

QKDSimulator

Simulator	QKDSimulator
Eigenschaften	• Online Simulator, mit dem QKD Protokolle analysiert und simuliert werden können • Parameter (Anzahl QBits, Fehlertoleranz etc…) können über Schieberegler eingestellt werden • Es stehen 4 Simulatoren-Typen zur Auswahl: *Complete QKD-Stack*, *Sifting*, *Biased Error Estimation* und *Shannon Bound* • Nach Simulation sind Ergebnisse unter eigenem Tab zusammengefasst • Ebenfalls sind Plots zu den Ergebnissen abrufbar
Einsatzgebiete	• Simulator für bestimmte Ausgangswerte zu QKD
Sprache	• Python-Programm
Lizenz	• K.A.
Zugang	• Keine Beschränkungen
OS	• Plattformunabhängig: Webbrowser erforderlich
Repository	• kein Respository
Dokumentation	• auf Webseite abrufbar

Quantum annealing

D-Wave Ocean

Simulator	D-Wave Ocean
Eigenschaften	• D-Wave Ocean ist eine Software Suite bestehend aus mehreren unabhängigen Paketen • Jedes Paket ist für einen individuellen Anwendungszweck konzipiert: So eignet sich beispielsweise dwave-hybrid zur Lösung von mathematischen Problemen auf hybriden (klassisch + basierend auf Quanten) Architekturen
Einsatzgebiete	• Lösung von schwer berechenbaren mathematischen Problemen. Je nach Softwarepaket kann der Code auch auf Quantenannealer der Firma D-Wave ausgeführt werden
Sprache	• Unterschiedliche Programmiersprachen. Abhängig von jeweiligem D-Wave Ocean Paket. Die meisten Pakete liegen in Python oder C++ vor
Lizenz	• Apache 2.0 Lizenz oder MIT Lizenz. Abhängig vom jeweiligem Softwarepaket
Zugang	• Keine Beschränkungen
OS	• Installation für Linux, Mac OS und Windows getestet
Repository	• Auf GitHub verfügbar
Dokumentation	• Dokumentationsseite zu D-Wave Ocean Softwareprodukten

Weitere Simulatoren

Weitere Simulatoren sind auf nachfolgenden URLs zu finden:

https://www.win-labor.dfn.de/quantentechnologien/quantensimulation/

https://quantiki.org/wiki/list-qc-simulators

https://github.com/qosf/awesome-quantum-software

https://qosf.org/project_list/