

# HHL - Algorithmus

Alfred Nguyen

Fakultät der Informatik  
Technische Universität München  
85758 Garching, Bavaria

June 2023

# Gliederung

Einführung

Mathematische Grundlagen

HHL Algorithmus

Einfaches Beispiel

Evaluierung

Zukunftsperspektiven

# Gliederung

Einführung

Mathematische Grundlagen

HHL Algorithmus

Einfaches Beispiel

Evaluierung

Zukunftsperspektiven

# Einführung

Wir haben schon viel über die wichtigsten Algorithmen gehört

- ▶ Shors-Algorithmus
- ▶ Grover-Algorithmus

Der HHL-Algorithmus

- ▶ erstellt von Aram Harrow, Avinatan Hassidim und Seth Lloyd
- ▶ lösen von sehr großen linearen Gleichungen

$$A\vec{x} = \vec{b}$$

# Motivation

Es löst grundlegendes Probleme in der Mathematik

- ▶ Least square fitting
- ▶ Optimierungs Probleme
- ▶ Simulationen und Imageprocessing
- ▶ ...

Kleine Revolution insbesondere bei Quantum Machine Learning

- ▶ HHL als Subroutine oder in erweiterten Form benutzt
- ▶ Approximation mit Computern braucht  $\min N$  Zeitschritte!

# Das Problem

## Gegeben:

- ▶  $A$  Matrix der Form  $n \times n$
- ▶  $\vec{b}$

## Löse das System:

$$A\vec{x} = \vec{b}$$

$$\vec{x} = A^{-1}\vec{b}$$

Wir sind also daran interessiert das Inverse  $A^{-1}$  zu finden

# Gliederung

Einführung

Mathematische Grundlagen

HHL Algorithmus

Einfaches Beispiel

Evaluierung

Zukunftsperspektiven

# Hermitsche Matrix

**Sei:**

- ▶  $A$  eine  $n \times n$  Matrix
- ▶  $A^T$  das transponierte von  $A$
- ▶  $\overline{A}$  das komplex konjugierter von  $A$
- ▶  $A^\dagger$  die Hermitsche Matrix von  $A$

**Dann:**

$$A = \overline{A^T} = A^\dagger$$



# Hermitsche Matrix

**Beispiel:**

$$A = \begin{bmatrix} 2 & 1 - i \\ 1 + i & 3 \end{bmatrix}$$

$$\overline{A} = \begin{bmatrix} 2 & 1 + i \\ 1 - i & 3 \end{bmatrix}$$

$$\overline{A^T} = \begin{bmatrix} 2 & 1 - i \\ 1 + i & 3 \end{bmatrix} = A = A^\dagger$$

Die Matrix  $A$  ist Hermitisch.

# Hermitsche Matrix

Falls eine Matrix  $A$  nicht Hermitisch ist:

$$A^\dagger = \begin{pmatrix} 0 & A \\ \overline{A^T} & 0 \end{pmatrix}$$

# Spektralzerlegung

**Gegeben:**

$$A = UDU^T$$

$$= \begin{bmatrix} U_1 & U_2 & \dots & U_n \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \lambda_n \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ \dots \\ U_n \end{bmatrix}$$

- ▶  $A$  eine  $n \times n$  Matrix
- ▶  $D$  ist eine Diagonalmatrix aus den Eigenwerten
- ▶  $U$  besteht aus den Eigenvektoren von  $A$

$A$  lässt sich aus seinen Eigenwerten und Eigenvektoren darstellen.

# Spektralzerlegung

Selbes gilt für das Inverse von  $A$

$$A^{-1} = U^T D^{-1} U$$

- ▶  $A^{-1}$  nur durch Eigenwerten und Eigenvektoren bestimmbar!
- ▶ Methode im klassischen nicht schneller
- ▶ für HHL Algorithmus sehr wichtig

# Veschränkung

Verschränkte Zustände können nicht durch einzelne Zustände dargestellt werden

$$|\Phi\rangle \neq |\phi\rangle |\psi\rangle$$

# Veschränkung

## Beispiel:

Nicht Verschränkt

$$\begin{aligned} |\Phi_1\rangle &= \frac{1}{\sqrt{2}}(|10\rangle + |11\rangle) \\ &= |1\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |1\rangle |+\rangle \end{aligned}$$

Verschränkt

$$\begin{aligned} |\Phi_2\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ &\neq |\alpha\rangle |\beta\rangle \end{aligned}$$

# Gliederung

Einführung

Mathematische Grundlagen

**HHL Algorithmus**

Einfaches Beispiel

Evaluierung

Zukunftsperspektiven

# Übersicht

Vergleich klassische zur quanten Version

Klassisch

$$A\vec{x} = \vec{b}$$

$$\vec{x} = A^{-1}\vec{b}$$

Quanten Version

$$A|x\rangle = |b\rangle$$

$$|x\rangle = A^{-1}|b\rangle$$

Spektralzerlegung von  $A$  und  $|b\rangle$

$$|x\rangle = A^{-1}|b\rangle = \sum_{i=0}^{2^{n_b}-1} \lambda_i^{-1} b_j |u_j\rangle$$

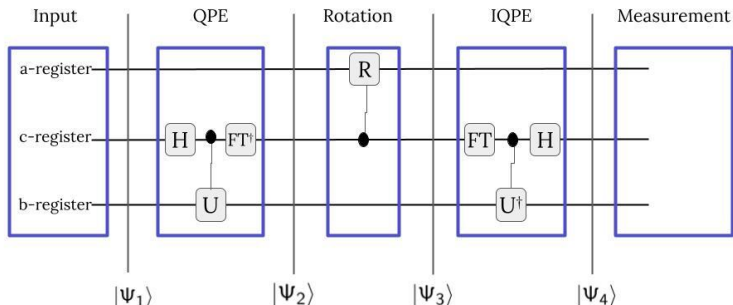


# Der Algorithmus

## Ablauf

1. State Preparation
  - ▶ Enkodiert Vektor und Matrix in Quanten Computer
2. Quantum Phase Estimation
  - ▶ ermittelt Eigenwerte
3. Ancilla Bit Rotation
  - ▶ Invertiert Eigenwerte
4. Inverse Quantum Phase Estimation
  - ▶ löst verschränkte Qubits auf
5. Messung
  - ▶ liest das Ergebnis  $|x\rangle$  aus

# Quantum Circuit



1. Ancilla (Helfer): a-register
  - ▶ Indikator qubit, zeigt ob Zustände verschränkt sind
2. Register: c-register
  - ▶ beinhaltet die Eigenwerte
3. Input: b-register
  - ▶ beinhaltet den Vektor  $\vec{b}$

# Gliederung

Einführung

Mathematische Grundlagen

HHL Algorithmus

**Einfaches Beispiel**

Evaluierung

Zukunftsperspektiven

## Einfaches Beispiel

Matrix  $A$  und Vektor  $\vec{b}$ :

$$A = \begin{pmatrix} 1 & -\frac{1}{3} \\ -\frac{1}{3} & 1 \end{pmatrix}$$

$$\vec{b} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Klassische Lösung

$$\vec{x} = \begin{pmatrix} \frac{3}{8} \\ \frac{9}{8} \end{pmatrix}$$

Verhältnis der Lösung:

$$\frac{|x_0|^2}{|x_1|^2} = \frac{\frac{9}{64}}{\frac{81}{64}} = \frac{1}{9}$$

# Einfach Beispiel

Eigenvektoren von  $A$  sind:

$$\vec{u}_0 = \begin{pmatrix} \frac{-1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix}$$

$$|u_0\rangle = \frac{-1}{\sqrt{2}} |0\rangle + \frac{-1}{\sqrt{2}} |1\rangle$$

$$\vec{u}_1 = \begin{pmatrix} \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$|u_1\rangle = \frac{-1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle$$

## Einfach Beispiel

Eigenvektoren enkodiert in die Quanten Version

$$\widetilde{\lambda}_j = N\lambda_j t / 2\pi$$

$$N = 4$$

$$t = 3\pi/4$$

Eigenvektoren von  $A$  sind:

$$\lambda_0 = \frac{2}{3}$$

$$\lambda_1 = \frac{4}{3}$$

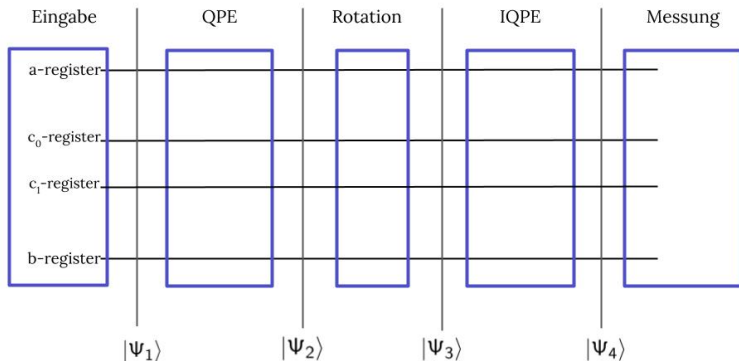
$$\widetilde{\lambda}_0 = \frac{4 * \frac{2}{3} * \frac{3\pi}{4}}{2\pi} = \frac{4 * 2 * 3\pi}{3 * 4 * 2\pi} = 1$$

$$|\widetilde{\lambda}_0\rangle = |01\rangle$$

$$\widetilde{\lambda}_1 = \frac{4 * \frac{4}{3} * \frac{3\pi}{4}}{2\pi} = \frac{4 * 4 * 3\pi}{3 * 4 * 2\pi} = 2$$

$$|\widetilde{\lambda}_1\rangle = |10\rangle$$

# State Preparation



1. Anzahl Qubit for a-register: 1
2. Anzahl Qubits für das c-Register:  $N = 2$
3. Anzahl Qubits für  $\vec{b}$ :  $n_b = \log_2(N) = \log_2(2) = 1$

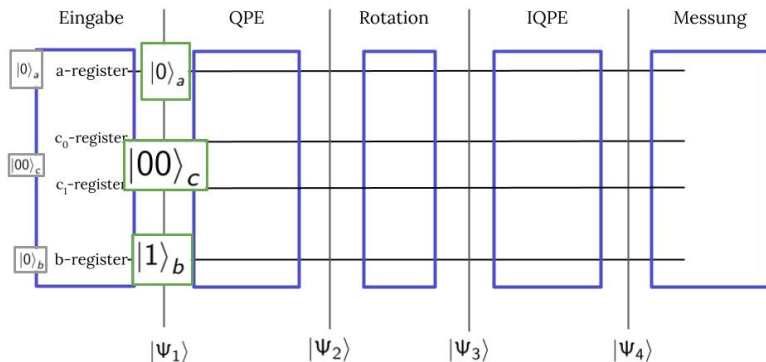
# State Preparation

- ▶  $\vec{b}$  wird als Quantenzustand  $|b\rangle$  kodiert
- ▶ in unserem Fall ist es sehr einfach

$$\vec{b} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \Leftrightarrow |b\rangle = 0|0\rangle + 1|1\rangle = |1\rangle$$



# State Preparation



Wir starten im 1 Zustand

$$|\psi_1\rangle = |1\rangle_b |00\rangle_c |0\rangle_a = |1000\rangle$$

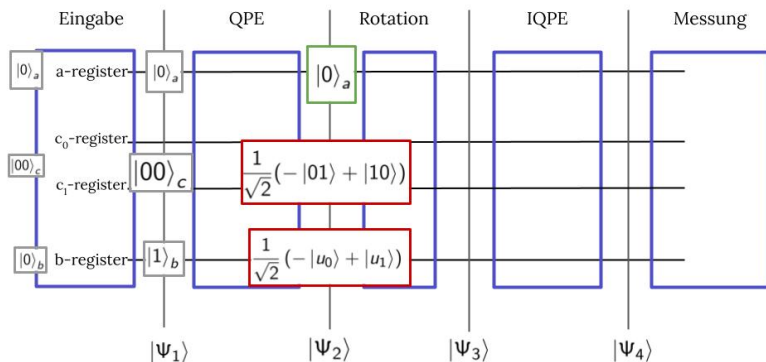
# Quantum Phase Estimation

Wir führen QPE aus:

$$\begin{aligned} |\psi_2\rangle &= |b\rangle_b |\tilde{\lambda}_j\rangle_c |0\rangle_a = \sum_{j=0}^{2^1-1} b_j |u_j\rangle |\tilde{\lambda}_j\rangle |0\rangle \\ &= \left( -\frac{1}{\sqrt{2}} |u_0\rangle |01\rangle + \frac{1}{\sqrt{2}} |u_1\rangle |10\rangle \right) |0\rangle \end{aligned}$$

- ▶ b-register: Zustand  $|b\rangle$  in Eigenbasis von A:  $|u_0\rangle$  or  $|u_1\rangle$
- ▶ jeweilige Koeffizienten:  $b_0 = \frac{-1}{\sqrt{2}}$  and  $b_1 = \frac{1}{\sqrt{2}}$
- ▶ c-register: Eigenwerte  $|\tilde{\lambda}_0\rangle$  und  $|\tilde{\lambda}_1\rangle$  enkodiert als  $|01\rangle$  und  $|10\rangle$
- ▶ a-register: ancilla Qubit  $|0\rangle$

# Quantum Phase Estimation



Wir erhalten:

$$|\Psi_2\rangle = \left( -\frac{1}{\sqrt{2}} |u_0\rangle |01\rangle + \frac{1}{\sqrt{2}} |u_1\rangle |10\rangle \right) |0\rangle_a$$

# Ancilla Rotation - Eigenwerte invertieren

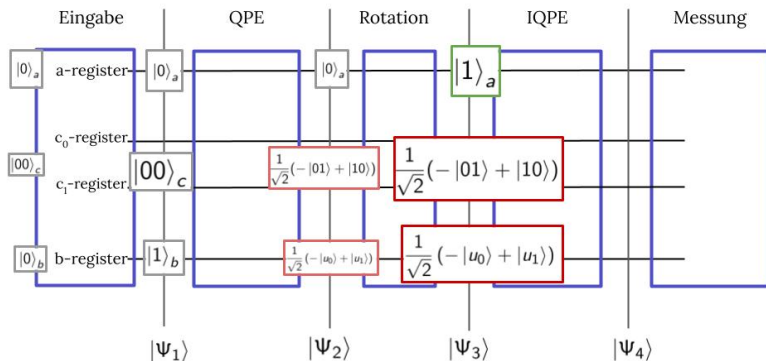
Wir invertieren das Ancilla Qubit:

$$\sum_{j=0}^{2^1-1} b_j |u_j\rangle |\tilde{\lambda}_j\rangle \left( \sqrt{1 - \frac{C^2}{\tilde{\lambda}_j^2}} |0\rangle + \frac{C}{\tilde{\lambda}_j} |1\rangle \right)$$
$$= \left( -\frac{1}{\sqrt{2}} |u_0\rangle |01\rangle (|0\rangle + |1\rangle) + \frac{1}{\sqrt{2}} |u_1\rangle |10\rangle \right) \left( \sqrt{1 - \frac{1}{4}} |0\rangle + \frac{1}{2} |1\rangle \right)$$

Wir gehen davon aus, dass wir  $|1\rangle$  messen.

$$= \sqrt{\frac{8}{5}} \left( -\frac{1}{\sqrt{2}} |u_0\rangle |01\rangle |1\rangle + \frac{1}{2\sqrt{2}} |u_1\rangle |10\rangle \right) |1\rangle_a$$

# Ancilla Rotation - Eigenwerte invertieren



$$|\Psi_3\rangle = \sqrt{\frac{8}{5}} \left( -\frac{1}{\sqrt{2}} |u_0\rangle |01\rangle |1\rangle + \frac{1}{2\sqrt{2}} |u_1\rangle |10\rangle \right) |1\rangle_a$$

# Inverse Quantum Phase Estimation

Wir führen IQPE aus:

$$|x\rangle_b |00\rangle_c |1\rangle_a$$

$$|x\rangle_b = A^{-1} |b\rangle = \sum_{i=0}^{2^1-1} \lambda_i^{-1} b_i |u_i\rangle$$

$$= \lambda_0^{-1} b_0 |u_0\rangle + \lambda_1^{-1} b_1 |u_1\rangle$$

$$= -\frac{1}{\frac{2}{3}\sqrt{2}} |u_0\rangle + \frac{1}{\frac{4}{3}\sqrt{2}} |u_1\rangle$$

$$= \frac{2}{3} \sqrt{\frac{8}{5}} \left( -\frac{1}{\frac{2}{3}\sqrt{2}} |u_0\rangle + \frac{1}{\frac{4}{3}\sqrt{2}} |u_1\rangle \right) |00\rangle_b |1\rangle_a$$

# Inverse Quantum Phase Estimation

Wegen Normalisierung der Eigenvektoren können wir

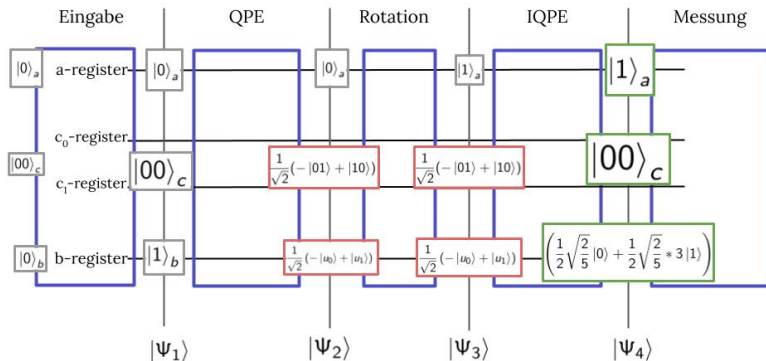
$$\blacktriangleright |u_0\rangle = \frac{-1}{\sqrt{2}} |0\rangle + \frac{-1}{\sqrt{2}} |1\rangle$$

$$\blacktriangleright |u_1\rangle = \frac{-1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle$$

$$|\Psi_4\rangle = \frac{1}{2} \sqrt{\frac{2}{5}} (|0\rangle + 3|1\rangle) |00\rangle_b |1\rangle_a$$

$$|\Psi_4\rangle = \left( \frac{1}{2} \sqrt{\frac{2}{5}} |0\rangle + \frac{1}{2} \sqrt{\frac{2}{5}} * 3 |1\rangle \right) |00\rangle_b |1\rangle_a$$

# Ancilla Rotation - Eigenwerte invertieren



$$|\psi_4\rangle = \left( \frac{1}{2} \sqrt{\frac{2}{5}} |0\rangle + \frac{1}{2} \sqrt{\frac{2}{5}} * 3 |1\rangle \right) |00\rangle_b |1\rangle_a$$



# Measurment

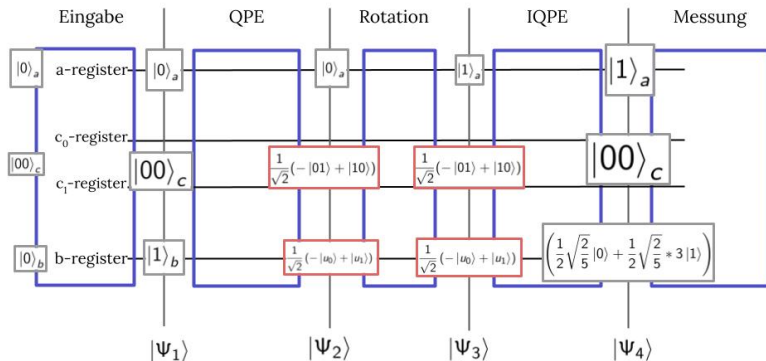
Um die Wahrscheinlichkeit von  $|u_0\rangle$  und  $|u_1\rangle$  zu erhalten, müssen wir ihre Koeffizienten quadrieren

$$c_0 = \left| \frac{1}{2} \sqrt{\frac{2}{5}} * 1 \right|^2 = \frac{1}{20}$$

$$c_1 = \left| \frac{1}{2} \sqrt{\frac{2}{5}} * 3 \right|^2 = \frac{9}{20}$$

Das Verhältnis im b-Register ist wie erwartet 1 : 9.

# Gesamte Rechnung



# Gliederung

Einführung

Mathematische Grundlagen

HHL Algorithmus

Einfaches Beispiel

**Evaluierung**

Zukunftsperspektiven

## Gauß Verfahren

$$\mathcal{O}(N^3)$$

- ▶ nicht der schnellste Algorithmus
- ▶ gleiche constraints sind zu beachten!!

## Klassisch

Conjugate gradient descent

$$\mathcal{O}(\kappa s \log\left(\frac{1}{\epsilon}\right) N)$$

- ▶  $N :=$  Anzahl an unbekannten
- ▶  $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$ : condition number

## Quanten Version

HHL

$$\mathcal{O}\left(\frac{\kappa^2 s^2}{\epsilon} \log N\right)$$

- ▶  $\epsilon :=$  Fehler des Ergebnisses
- ▶  $s :=$  is s-sparse Matrix: jede Zeile hat max. s Einträge

# Laufzeit

## Klassisch

Conjugate gradient descent

$$\mathcal{O}\left(\kappa s \log\left(\frac{1}{\epsilon}\right) N\right)$$

$$\Rightarrow \mathcal{O}(N)$$

## Quanten Version

HHL

$$\mathcal{O}\left(\frac{\kappa^2 s^2}{\epsilon} \log N\right)$$

$$\Rightarrow \mathcal{O}(\log(N))$$

## Takeaway

- ▶ exponentialer speed up  $\mathcal{O}(N)$  vs  $\mathcal{O}(\log(N))$
- ▶ klassischer algorithmus hat bessere Fehlerabhängigkeit:  
 $\log\left(\frac{1}{\epsilon}\right)$  vs  $\frac{1}{\epsilon}$

# Einschränkungen

1. einfache Zustandsvorbereitung des Vektors  $\vec{b}$  zum Quantenzustand  $|b\rangle$
2. niedrige condition number  $\kappa$
3.  $A$  muss s-sparse sein
4. nicht jeder Eintrag von  $|x\rangle$  auslesbar
5. Der Ressourcenbedarf sehr hoch

# Einschränkungen

1. niedrige condition number (es ist außerdem nicht einfach  $\kappa$  im Vorhinein zu ermitteln)
2. muss s-sparse sein
3. einfache Zustandsvorbereitung des Vektors  $\vec{b}$  zum Quantenzustand  $|b\rangle$ 
  - ▶ wenn man  $|b\rangle$  klassisch lesen/schreiben muss, ist der Geschwindigkeitsgewinn weg, da  $|b\rangle$   $N$  Einträge hat  $\rightarrow$  qram
4. nicht jeder Eintrag von  $|x\rangle$  auslesbar
  - ▶ Nachbearbeitung muss erfolgen
  - ▶ nur  $\log_2(n)$  Qubits -  $\rightarrow$  nur eine Näherung
  - ▶ statistische Informationen möglich (Verhältnis, Bereiche großer Einträge, ...)
5. Der Ressourcenbedarf sehr hoch
  - ▶ Shors Algorithmus ist dem HHL-Algorithmus sehr ähnlich (aufgrund von QPE)
  - ▶ untere Grenze von 4000 logischen Qubits (2048bit RSA)
  - ▶ d.h. millionen physikalischer Qubits (für Fehlerkorrektur)



# Gliederung

Einführung

Mathematische Grundlagen

HHL Algorithmus

Einfaches Beispiel

Evaluierung

Zukunftsperspektiven

# Anwendungen

## Hauptproblem

- ▶ Hauptproblem: gibt keinen vollständigen Vektor aus
- ▶ Aber einige Probleme können mit dieser Methode gelöst werden:

# Anwendungen

## Machine Learning: Least-Square-Fitting

- ▶ Datenanpassung mit Least Square Fitting
- ▶ durch Berechnung einer Schätzung der inversen Matrix

## Analysis of Large Sparse Electrical Networks

- ▶ Elektrizitätsnetz vielen verbundenen Komponenten
- ▶ geringe Anzahl Verbindungen zwischen den Komponenten
- ▶ Berechnung des Widerstands durch approximation von Erwartungswerten

Es wäre wichtig, mehr Anwendungen zu finden, welche den Anforderungen entsprechen.

# Anwendung in IT-Security

## HHL in der IT-Security

- ▶ in erster Linie nur für Lösen von linearen Systemen
- ▶ nicht direkt mit IT-Security verbunden
- ▶ aber Potenzial als Subroutine angewendet zu werden

## Mögliche Anwendungen

- ▶ secure multi-party computation
- ▶ zero-knowledge proofs
- ▶ cryptographic key generation and management
- ▶ big data analysis/pattern recognition (für Betrugserkennung)

# Variationen

## Modifikationen und Optimierung

- ▶ QRAM zur Vorbereitung von  $|b\rangle$
- ▶ kein Ancilla-Bit erforderlich unter bestimmten Voraussetzungen
- ▶ Variable time amplitude amplification um condition number  $\kappa$  zu verbessern

# Perspektive

- ▶ Großer Einfluss im Bereich Quantum Machine Learning
- ▶ noch keine bahnbrechenden Anwendungen (wie z.B. Shors Algorithmus zum Brechen von RSA)
- ▶ aber viel aktive Forschung um neue Verbesserungen im Algorithmus zu finden
- ▶ zeigt deutlichen Fortschritt in der Quantencomputing Welt