Lineare Gleichungssysteme am Quantencomputer

Sei $A \in \mathbb{C}^{N \times N}$, $b \in \mathbb{C}^N$, $N = 2^n$

(invertierbar, hermitesch (komp. symm.))

<u>LGS</u>: finde $x \in \mathbb{C}^N$ s.t. $Ax = b$

für A pos. def., sparse $\rightsquigarrow$ klass. Verfahren Standard CG

Aufwand $\mathcal{O}(sN \cdot \ell)$ s ... sparsity, ℓ ... Anzahl Iterationen

Fehler $\|x - x_\ell\|_A \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^\ell \|x - x_0\|_A$

$\kappa = \kappa(A)$ ... Konditionszahl $= \|A\| \cdot \|A^{-1}\|$

für rel. Fehler $\leq \varepsilon$

$$\Rightarrow 2\underbrace{\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)}^{\ell} \leq \varepsilon$$

$$q := \left(1 - \frac{2}{\sqrt{\kappa}+1}\right) \geq e^{-\frac{2}{\sqrt{\kappa}}}$$

$$\Rightarrow \ell \leq \frac{\log \frac{\varepsilon}{2}}{\log q} \leq \frac{1}{2}\sqrt{\kappa} \ln\left(\frac{2}{\varepsilon}\right)$$

$\rightsquigarrow$ Aufwand $\mathcal{O}\left(sN\sqrt{\kappa} \ln \frac{2}{\varepsilon}\right)$

Q: exponentieller Speedup auf Q-Comp. möglich??

Quantenversion von LGS:

QLGS: $A \in \mathbb{C}^{N \times N}$ hermitesch, $\det A = 1$

$(b_i) = b \in \mathbb{C}^N$, $x \overset{=(x_i)}{\in} \mathbb{C}^N$ s.t. $x = A^{-1} b$

gegeben, $N = 2^n$.

Sei weiters $|b\rangle$ ein n-qubit Zustand gegeben als

$$|b\rangle := \sum_i b_i |i\rangle / \| \sum_i b_i |i\rangle \|$$

und

$$|x\rangle := \sum_i x_i |i\rangle / \| \sum_i x_i |i\rangle \|$$

Ziel: finde Zustand $|\tilde{x}\rangle$ s.d.

$$\| |x\rangle - |\tilde{x}\rangle \| \leq \varepsilon$$ ← geg. Fehlertoleranz

mit Wahrsch. $\Omega \geq \frac{1}{2}$

Schreibweise in Literatur (formal!)

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

Bem.: ·) Normalisierungen notwendig da sonst kein Q-Zustand!

·) QLSP ≠ LSP
↳ man erhält nur Zustand $|x\rangle$, nicht Vektor, der $Ax=b$ löst!

·) Falls A nicht hermitsch betrachte

$$\tilde{A} = \begin{pmatrix} 0 & A^H \\ A & 0 \end{pmatrix} = |1\rangle\langle 0| \otimes A + |0\rangle\langle 1| \otimes A^H$$

⇝ Matrix dim. verdoppelt
≃ 1 zusätzliches Ancilla Qubit

löse $\tilde{A}|0x\rangle = |1b\rangle$

·) $\det A = 1$ keine echte Einschränkung
⇝ skalierte Matrix

·) wollen: effizienten Algor.
↳ poly logarithmisch in N

·) impl. Annahme: Zustand $|b\rangle$ kann effizient bereitgestellt werden
A kann am Q-Comp. implementiert werden
↳ später!

Anm. Auslesen von Koeff. von $|x\rangle \rightsquigarrow \mathcal{O}(N)$

$\rightarrow$ QLSP nur nützlich wenn Quantenzust. benötigt

$\rightsquigarrow$ Subroutine für andere Probleme

HHL (Harrow, Hassidim, Lloyd) - Algorithmus

Idee: A hermitsch $\rightarrow$ Spektralsatz:

$$A = \sum_{j=0}^{N-1} \lambda_j |u_j\rangle\langle u_j|$$

$\lambda_j \in \mathbb{R}$ -- EW $\quad |u_j\rangle \in \mathbb{R}^N$ -- EV

$$|b\rangle = \sum_{j=0}^{N-1} \beta_j |u_j\rangle = \sum_{j=0}^{N-1} \langle u_j|b\rangle |u_j\rangle$$

Spektralsatz (oder direkt nachrechnen):

$$A^{-1} = \sum_{j=0}^{N-1} \frac{1}{\lambda_j} |u_j\rangle\langle u_j|$$

$$\Rightarrow |x\rangle = A^{-1}|b\rangle = \sum_{j=0}^{N-1} \frac{\beta_j}{\lambda_j} |u_j\rangle$$

nur formal zu verstehen $\rightsquigarrow$ als korrekte Quantenoperation später!

<u>Problem</u>: A, A^{-1} i.A. nicht unitär

$\Rightarrow A^{-1}|b\rangle$ nicht erlaubt

<u>Lösung</u>: Matrix $U = e^{iA}$ unitär,

hat selbe EV wie A da

$$A = XDX^{-1} \Rightarrow e^{iA} = \sum_{n=0}^{\infty} \frac{(iA)^n}{n!}$$

$$= \sum_{n=0}^{\infty} \frac{(iXDX^{-1})^n}{n!}$$

$$= X \sum_{n=0}^{\infty} \frac{(iD)^n}{n!} X^{-1}$$

$$= X e^{iD} X^{-1}$$

$$= \sum_j e^{i\lambda_j} |u_j\rangle\langle u_j|$$

---

1) berechne EW von U

$\Rightarrow$ EW von A

$\hookrightarrow$ mittels Quantum Phase Estimation!

2) Invertiere EW $\lambda_j \mapsto \frac{1}{\lambda_j}$

wie ?? $\rightsquigarrow$ „controlled rotation"

3) Reversiere QPE

Hamiltonian Simulation

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

Q: Wie kann man $e^{iAt} = U$ bzw. U^{2^k} effizient implementieren?

hier: A ist Hamiltonian für U

obige Frage ist eine fundamentale Fragestellung in Quantencomputing!

Schrödinger Glg.s

$$i \frac{d}{dt} |\psi(t)\rangle = H |\psi(t)\rangle$$

beschreiben jedes Q-System

$$\hookrightarrow |\psi(t)\rangle = e^{-iHt} |\psi(0)\rangle$$

zur eff. Simulation $\rightsquigarrow$ muss e^{-iHt} eff. impl.!

„Hamiltonian Simulation"

als circuit bis auf Fehler $\leq \varepsilon$

Definition Ein Hamiltonian H, der auf n qubits agiert kann effizient simuliert werden, wenn

$\forall t > 0, \varepsilon > 0$ $\exists$ q-circuit U_H bestehend aus $\mathrm{poly}(n, t, \frac{1}{\varepsilon})$ gates s.d.

$$\| U_H - e^{-iHt} \| < \varepsilon$$

Bem. Zeitabh. wichtig
„no-fast forwarding theorem"
generell: min. Zeit zur Simulation von H an t
$\rightsquigarrow \Theta(t)$

---

generelles approx. Problem $\rightsquigarrow$ NP schwer Gate Zerl. zu finden

$\rightarrow$ brauchen Annahmen

Ann. 1: $H = \sum_{j=1}^{m} H_j$ $\quad m \sim \mathrm{poly}(n)$

& alle H_j agieren auf $k = \Theta(1)$ qubits

„k-local Hamiltonians"

Trotter-Suzuki splitting

Für k-local Ham. Implementierung von

$e^{iH_j t}$ einfacher effizient als H_j nur auf k qubits agiert

falls z.B. H diagon. $\Rightarrow H = T D T^H$

$$\Rightarrow e^{iHt} = T e^{iD} T^H$$

falls D_{ee} effizient bestimmbar

$$\Rightarrow |i0\rangle \xrightarrow[\text{Coordinates}]{} |iD_{ee}\rangle \xrightarrow[\text{Phase gate}]{} e^{iD_{ee}t} |iD_{ee}\rangle$$

$$\xrightarrow[\text{de Coord entry}]{} e^{iD_{ee}t} |i0\rangle$$

für k-local $\rightsquigarrow$ effiziente Diagonalisierung der H_j ✓

<u>Aber:</u> $e^{iHt} \neq \prod_{j=1}^{m} e^{iH_j t}$

gilt nur, wenn alle H_j kommutieren!

da $e^{A+B} = \sum_{n=0}^{\infty} \frac{(A+B)^n}{n!} = \sum_{n=0}^{\infty} \sum_{k=0}^{n} \binom{n}{k} \frac{A^k B^{n-k}}{n!}$ $\quad = \frac{1}{k!(n-k)!}$

$$\overset{\text{Cauchy prod.}}{\underset{\text{nur für Komm!}}{=}} \sum_{k=0}^{\infty} \frac{A^k}{k!} \cdot \sum_{n=0}^{\infty} \frac{B^n}{n!} = e^A \cdot e^B$$

Lösung: Trotter / Lie-Produkt Formel

Theorem Sei $H = H_1 + H_2$, alle hermitsch

$$\Rightarrow \lim_{\ell \to \infty} \left(e^{iH_1 t/\ell} \cdot e^{iH_2 t/\ell} \right)^{\ell} = e^{iHt}$$

Es gilt sogar

$$\left\| e^{iHt} - \left(e^{iH_1 t/L} \cdot e^{iH_2 t/L} \right)^{L} \right\| \leq C \cdot \frac{t^2}{L}$$

mit $C = C(\|H_1\|, \|H_2\|)$

Beweis Taylor:

zu verstehen als Terme in H_1, t, L beschr. durch

$$e^{iH_1 t/L} = I + iH_1 \frac{t}{L} + \mathcal{O}\left(\|H_1\|^2 \frac{t^2}{L^2} \right)$$

$$\Rightarrow e^{iH_1 t/L} \cdot e^{iH_2 t/L} = \left(I + iH_1 \frac{t}{L} + \mathcal{O}\left(\|H_1\|^2 \frac{t^2}{L^2} \right) \right) \cdot \left(I + iH_2 \frac{t}{L} + \mathcal{O}\left(\|H_2\|^2 \frac{t^2}{L^2} \right) \right)^{L}$$

$$= \left(I + i \cdot (H_1 + H_2) \frac{t}{L} + \mathcal{O}\left(\max(\|H_1\|, \|H_2\|)^2 \cdot t^2/L^2 \right) \right)^{L}$$

$$(*) \quad \overset{\text{Taylor}}{=} \Big(\underbrace{e^{i(H_1+H_2)\frac{t}{L}}}_{=:A} + \underbrace{\mathcal{O}\left(\max(\|H_1\|, \|H_2\|)^2 \frac{t^2}{L^2} \right)}_{=:B} \Big)^{L}$$

Da $(A+B)^L = A^L + \sum_{j=0}^{L-1} A^{L-j-1} B A^j + \underbrace{\dots + B^L}_{\mathcal{O}(\|B\|^2)}$

$$\Rightarrow (*) = e^{i(H_1+H_2)t} + L \cdot \mathcal{O}\left(\max(\|H_1\|, \|H_2\|)^2 \frac{t^2}{L^2}\right)$$

$$\Rightarrow \left\| e^{i(H_1+H_2)t} - \left(e^{iH_1 t/L} \cdot e^{iH_2 t/L}\right)^L \right\| \leq C \frac{t^2}{L} \xrightarrow{L \to \infty} 0$$

$\square$

<u>Bem.</u> ·) Für eff. Simulation:

$$\max(\|H_1\|, \|H\|_2) = \mathcal{O}(\mathrm{poly}(n))$$

·) Fehler $\leq \varepsilon$ ...geg.

$$\Rightarrow L = \mathcal{O}\left(C(H_1, H_2), \frac{t^2}{\varepsilon}\right)$$

·) hier: Splitting 1. Ordnung

höhere Ordnung auch möglich, z.B. Strang splitting

$$\left\| e^{iHt} - \left(e^{iH_2 t/(2L)} e^{iH_1 t/L} e^{iH_2 t/(2L)}\right)^L \right\| \leq C \frac{t^3}{L^2}$$

$$\Rightarrow L = \mathcal{O}\left(C \frac{t^{\frac{3}{2}}}{\sqrt{\varepsilon}}\right)$$

Strang splitting: 2. Ordnung

Methoden Ordnung p möglich

$$\rightsquigarrow L = \mathcal{O}\left(t^{\frac{p+1}{p}} \varepsilon^{-\frac{1}{p}} \right) \quad \leftarrow \text{fast linear in } t!$$

·) Falls k-local Ham. mit m-Termen:

$$\left\| e^{iHt} - \left(e^{iH_1 t/L} \cdot \ldots \cdot e^{iH_m t/L} \right)^L \right\| \leq C \frac{m^2 t^2}{L}$$

·) Fehler Abschätzung rel. pessimistisch, in Anw. besser

Vorteil: einfach, braucht keine zus. Qubits

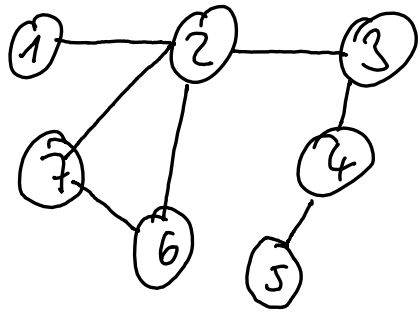
Essentielle Frage: Wie findet man Zerlegung

$$H = \sum H_i \quad ??$$

z.B. mittels Graphentheorie „graph coloring methods"

Def. Ein ungerichteter Graph $G=(V,E)$ ist def. durch Knotenmenge V und Kanten E, die ungeordnete Paare von Knoten sind.

Bsp.



$V=\{1,\dots,7\}$

$E=\{(1,2),(2,3),(2,6),(2,7),(6,7),(3,4),(4,5)\}$

Def. $v_1, v_2 \in V$ heißen <u>verbunden</u>, wenn $\exists\, e \in E$ s.d. $e=(v_1,v_2)$.

Der <u>Grad</u> eines Knotens $v \in V$ ist $\#\{v_i : (v,v_i) \in E\}$.

Sei $|V|=n$. Dann kann ein Graph mittels seiner $n \times n$-<u>adjecency matrix</u> A beschrieben werden:

$$A_{ij} = \begin{cases} 1 & \text{falls } i,j \in V : (i,j) \in E \\ 0 & \text{sonst} \end{cases}$$

<u>obiges Bsp</u>:

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

<u>graph coloring problem</u>: gegeben k Farben, können alle Kanten E von G so eingefärbt werden, dass zwei Kanten, die einen Endpunkt teilen nie die gleiche Farbe haben?

<u>Antwort</u>: Vizings Theorem:

<u>Theorem</u> Sei $G=(V,E)$ ein Graph mit maximalem Grad der Knoten d

$\Rightarrow \exists$ Lösg. für das gc-problem mit $k \leq d+1$

Anwendung auf Ham. Sim.:

1. Identifiziere H mit adj. matrix A:

$$A_{ij} = \begin{cases} 1 & \text{falls } H_{ij} \neq 0 \\ 0 & \text{falls } H_{ij} = 0 \end{cases}$$

$\rightsquigarrow$ liefert zugehörigen Graph G

2. Bestimme eine Einfärbung von G mit k Farben

3. Zerlege G bzw. A anhand der Farben in Sub-Graphen

$$A = \sum_{c=1}^{k} A_c$$

Für allg. A $\rightsquigarrow$ unmöglich effizient realisierbar!

$\rightsquigarrow$ verlange <u>sparsity</u>

<u>Annahme</u> $A \in \mathbb{C}^{2^n \times 2^n}$ Hermitsch, $\|A\| \leq 1$ sei <u>s-sparse</u> und wir haben <u>sparse access</u> auf die Matrix-Einträge. D.h.:

- jede Zeile/Spalte von A hat max. s <u>nicht-0 Einträge</u>
- Wir haben <u>query</u>

$$O_A : |i\,j\rangle|0\rangle \mapsto |i\,j\rangle|A_{ij}\rangle$$

hier: hinteres Register groß genug, dass $A_{ij} \in \mathbb{C}$ exakt/hinreichend genau als Binärdarst. geschr.

- Wir haben weitere <u>query</u>

$$O_C : |j\,l\rangle \mapsto |j\,r(j,l)\rangle$$

wobei $r(j,l) \in \{0, \dots, N-1\}$ die Position der l-ten nicht-0 Eintrag in der j-ten Spalte von A ist.

- Wir können O_A^{-1}, O_C^{-1} ausführen.

- Sparsity $\rightsquigarrow$ max. Grad der Graphen $\leq s$
- Vizing Thm $\rightsquigarrow$ brauche max. $s+1$ Farben
- Effizient berechenbare Einfärbung mit s^2 Farben
- Matrizen A_C sind symmetrisch, 1-sparse
  $\Rightarrow$ zugeh. H_C effizient simulierbar da Matrizen in $O(1)$ diagonalisierbar

  o.B.d.A. Diagonale immer 0 da $H = \underbrace{\mathrm{diag}(H)}_{\text{effizient } \checkmark} + \text{Rest}$
- Gesamtaufwand (naive Realisierung hier)

$$O(s^2 t^2 \mathrm{poly}(n)/\varepsilon)$$

Q: Wie wird „sparse access" implementiert?

Bsp. circulant matrix

$$\begin{pmatrix} \alpha & \gamma & 0 & \cdots & 0 & \beta \\ \beta & \ddots & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \ddots & \gamma \\ 0 & & & \ddots & \ddots & \beta \\ \gamma & & & & \beta & \alpha \end{pmatrix}$$

3-sparse

$\cdot)\ r(j,l) = j+l-1 \bmod N \quad l = 0,1,2$

(hier Matrizen von 0 weg indiziert)

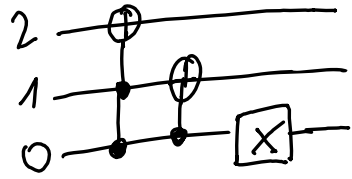
$$\rightarrow O_C: \ |j\rangle \mapsto \begin{cases} |\mathrm{mod}(j-1,N)\rangle & l=0 \\ |j\rangle & l=1 \\ |\mathrm{mod}(j+1,N)\rangle & l=2 \end{cases}$$

$\mathrm{mod}(j\pm 1,N)$ können durch shift Permutationen realisiert werden

$$R = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ \vdots & & 1 & \\ 0 & & & \ddots \\ 1 & \cdots & & 0 \end{pmatrix} \quad L = \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 1 & & & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & & \\ 0 & & 1 & 0 \end{pmatrix}$$

Circuits:

$\boxed{L}$:



(für 3 Qubits)

$\left(X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \text{bitflip}\right)$

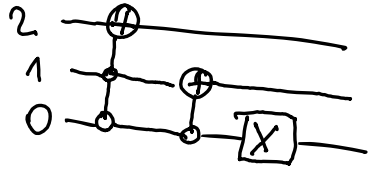
zu verstehen als Stellen in Binärdarst.

z.B.: $|010\rangle \mapsto |011\rangle$ also $2 \mapsto 3$

$\uparrow 2 \quad \uparrow 1 \quad \uparrow 0 \qquad \uparrow 2 \quad \uparrow 1 \quad \uparrow 0$

$|111\rangle \mapsto |000\rangle$ $\quad 7 \to 0$

$\boxed{R}$:



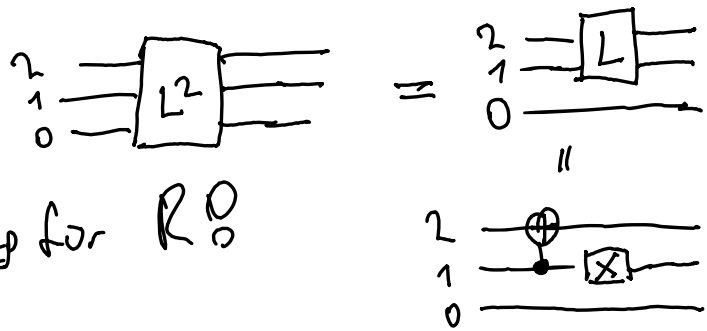
hier  controlled NOT, aktiv wenn Kontrolle = 0

 analog CCNOT, aktiv wenn beide = 0

---

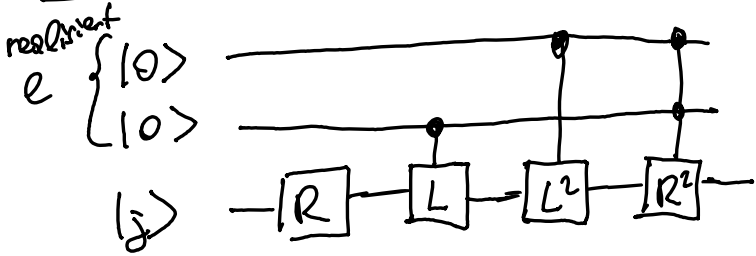
brauchen auch $\boxed{L^2}$, $\boxed{R^2}$

es gilt:



analog für R!

O_C circuit:



- R-gate für <u>alle</u> ℓ $\quad j \mapsto \mathrm{mod}(j-1, N)$

  passt für $\ell = 0$

  also $|00\rangle$

- für $\ell = 1 \simeq |01\rangle$ $\quad j \mapsto j$

  $\rightsquigarrow$ mache R-gate mittels L-gate rückgängig

- $\ell = 2 \simeq |10\rangle$ $\quad j \mapsto \mathrm{mod}(j+1, N)$

  mache L^2-shift $\Rightarrow RL^2 = L$-shift

- $\ell = 3 \simeq |11\rangle$ ... entspricht 0-Einträgen

  $\rightsquigarrow$ mache R^2-shift um L^2-shift umzukehren

O_A Circuit

generell: <u>controlled rotations</u>: unitäre Op.

$$U_\theta : |\theta\rangle |0\rangle \mapsto |\theta\rangle \left(\cos(\pi\theta)\, |0\rangle + \sin(\pi\theta)\, |1\rangle \right)$$

Kontrolle Ziel

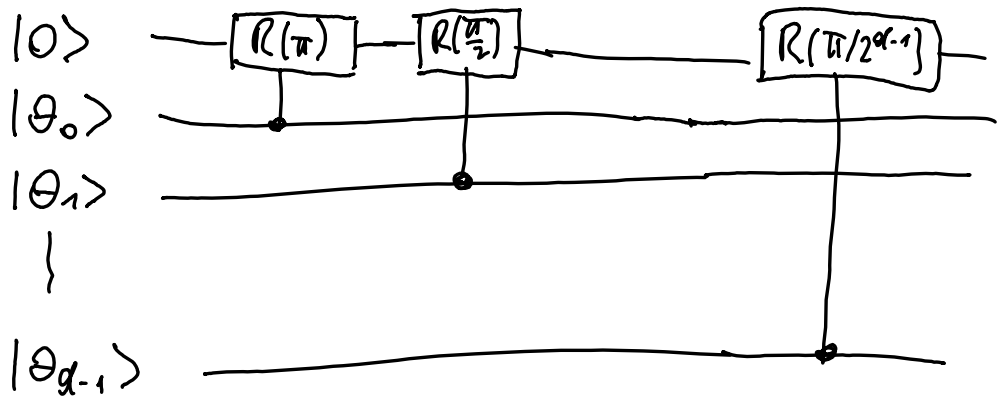
<u>hier</u>: $\theta \in [-1,1]$, <u>Ann.</u> $\theta = \theta_0 \cdot 2^{-1} + \theta_1 \cdot 2^{-2} + \dots + \theta_{d-1} \cdot 2$
exakte Binärdarst.

für $|\theta\rangle = |0\rangle \rightsquigarrow$ Output $|0\rangle|0\rangle$
$|\theta\rangle = |1\rangle \rightsquigarrow$ Drehung um $\pi\theta$ $\rightarrow$ Matrix $\begin{pmatrix} \cos J & -\sin J \\ \sin J & \cos J \end{pmatrix}$
$\overset{!}{=} R(2J)$

Schreibweise 

$\simeq$ 1-qubit Rotation um y-Achse

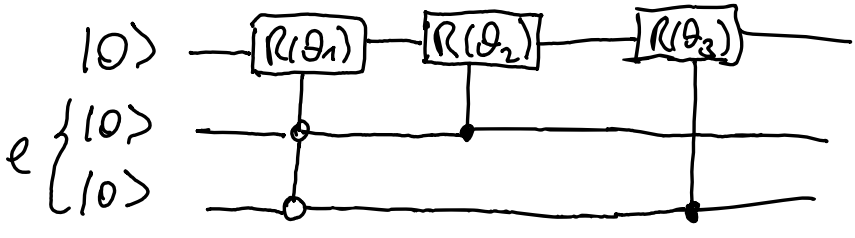
Implementierung



$\rightsquigarrow \mathcal{O}(d)$-gates

$\rightsquigarrow$ generiere Wert $\alpha^{\in[0,1]}$ mittels Rotation um Winkel $\theta_1 = \arccos\alpha \cdot \frac{1}{\pi}$

analog $\rightsquigarrow \theta_2 = \frac{1}{\pi}\arccos(\beta)$, $\theta_3 = \frac{1}{\pi}\arccos(\gamma)$



$\ell = 0 \rightsquigarrow \alpha$, $\ell = 1 \rightsquigarrow \beta$, $\ell = 2 \rightsquigarrow \gamma$

Modifikation für 3-diag Matrix:

füge 2 zusätzliche Rotationen $R(\theta_4), R(\theta_5)$ hinzu, die Elemente A_{1n} bzw A_{n1} auf 0 zurückrotieren

$\rightsquigarrow$ brauche 1 control Qubit mehr

Einlesen der rechten Seite

gegeben $b \in \mathbb{R}^N$, Ziel: $|b\rangle := \frac{\sum b_i |i\rangle}{\|b\|_2}$

in $\mathcal{O}(\log N)$ (bzw $|\tilde{b}\rangle$ mit $\tilde{b} \approx b$ bis auf machine prec.)

mögliche Lösg: <u>quantum RAM</u> (qRAM)

klassische Datenstruktur auf die mit Überlagerungen von q-Zuständen zugegriffen werden kann.

---

oBdA $\|b\|_2 = 1$ da das feste Zahl

$\rightsquigarrow$ kann mittels Binärdarst. eingelesen werden

---

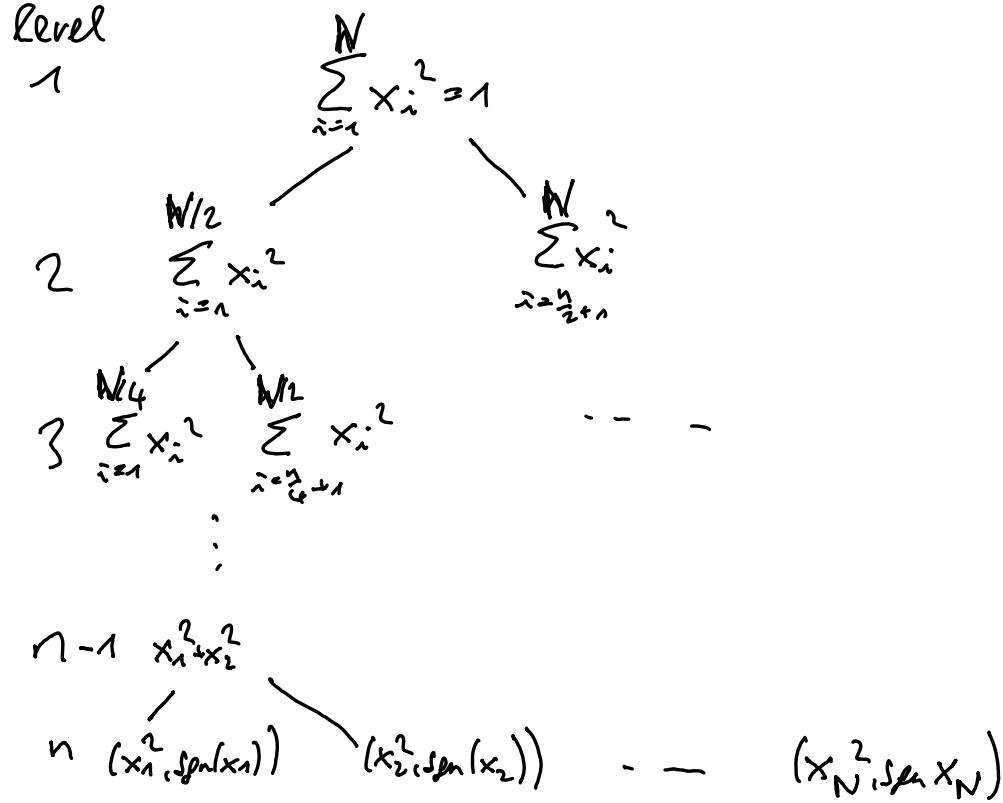
divide and conquer – Idee $\rightsquigarrow$ speichere

x in Binärbaum B_x

Wurzel $\sum_{i=1}^{N} x_i^2 = 1$

Tiefe $\log N = n$

Blätter $(x_i^2, \operatorname{sgn} x_i)$



<u>Anm.</u> ·) Jeder Knoten ist Summe der Kinder
(zusätzliche sgn. Information im Blatt ignoriert)

·) Vektor wird geladen: Läuft durch Baum von Wurzel weg, fügt wenn nötig <u>neue Register hinzu</u>, verwendet <u>controlled rotation</u>, um Werte in den jeweiligen Nodes zu realisieren

·) B_x hat $\mathcal{O}(N)$ Knoten

<u>Ann.</u>: ist pre-computed (klassisch)
↳ isomorphic in Aufwand...

<u>Algorithmus</u> (lade Vektor aus qRAM)

% input: $x \in \mathbb{R}^N$ gesp. im Baum B_x
% output: $|x\rangle$

Initialisiere n-qubits $|0 \ldots 0\rangle$, bez. mit $|q_1\rangle \ldots |q_n\rangle$

$r = \text{root}(B_x)$, call processNode(r)

processNode (vertex u)

u_l, u_r -- Kinder(u)

$\theta = \arccos \sqrt{\frac{u_l}{u}}$

Controlled Rotation $|q_k\rangle = \cos\theta\,|0\rangle + \sin\theta\,|1\rangle$

falls qubits $|q_1\rangle \ldots |q_{k-1}\rangle \simeq$ Binärdarst. von u

if u_l, u_r Blätter

$|q_k\rangle =$ processSign (q_k, u_l, u_r)

return

else

processNode (u_l)

processNode (u_r)

end

end

```
procedure Sign ( q_k, v_l, v_r )
    if sgn(v_l) = sgn(v_r) = +1     return q_k
    if  —  ||  = -1                 return -q_k
    if sgn(v_l) = 1, sgn(v_r) = -1  return Z q_k
    if sgn(v_l) = -1, sgn(v_r) = 1  return -Z q_k
end
```

Z-phase flip

<u>Lemma</u> Falls $x \in \mathbb{R}^N$ in B_x vorgespeichert

$\Rightarrow$ Alger. erzeugt Zustand $|x\rangle = \sum_{i=1}^{N} x_i |i\rangle$

in $\mathcal{O}(n) = \mathcal{O}(\log N)$.

<u>Beweis</u>.) Zeige Alg. erzeugt tatsächlich $|x\rangle$

Alg. $\rightsquigarrow$ wandere durch Baum, multipliziere

Knoten v_k mit $\sqrt{\frac{v_k}{v_{k-1}}}$, Blatt i noch mit signum $\quad k \simeq$ level

$$\Rightarrow \prod_{k=2}^{n} \sqrt{\frac{v_k}{v_{k-1}}}\, \mathrm{sgn}(x_i) = \sqrt{\underbrace{\frac{v_n}{v_1}}_{\frac{x_i^2}{1}}}\, \mathrm{sgn}(x_i) = x_i \quad \checkmark$$

<u>Laufzeit</u>: 2^k Rotationen (in $\mathcal{O}(1)$) auf level k

passiert parallel $\rightsquigarrow$ kontrollierte Op.

auf selben Qubit !

(Kontrolle checkt binär-Repr. von Vorfahre)

$\rightsquigarrow$ # Levels $= n = \log N$ beschr. Aufwand

<u>Aber</u>: Annahme B_x precomputed signifikant!

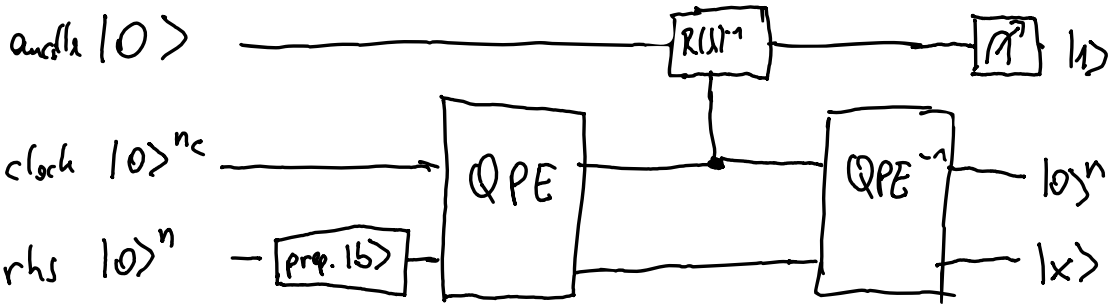
kann Speedup ruinieren!

Zumindest: Lemma zeigt: kann dann $|b\rangle$ und

auch Kopien von $|b\rangle$ $\binom{0}{0}$ schnell erzeugen

HHL revisited

Circuit



1. bef. $|b\rangle$ (z.B. QRAM)
   mit $n = \log N$ qubits um $|b\rangle$ darzustellen

2. Wende Quantum Phase Estimation an auf

$$|0\rangle\,|b\rangle = \sum_{j=0}^{N-1} \beta_j\, |0\rangle |u_j\rangle \qquad (\text{hier } |0\rangle = |0\rangle^{n_c})$$

mit $U = e^{iA}$

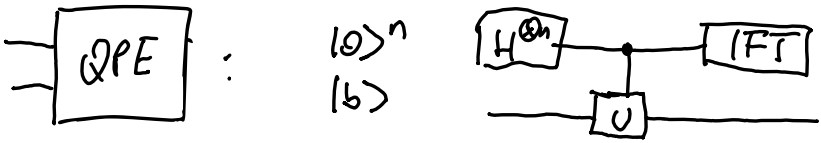
benötigt: $H^{\otimes n_c}$ ✓

U^{2^j} $\quad j = 0, \dots, n_c - 1$ Ham. Sim. ✓

IQFT $\to \mathcal{O}(n^2)$-gates ✓

liefert: Approx. zu EWen von A (nicht von U → gut)

Approx., da in QPE angenommen: λ_j hat <u>exakte</u> Binärdarst. mit n_c-qubits



realisiert $|j\rangle|\psi\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i \lambda j} |j\rangle|\psi\rangle$

mit λ EW zu EV ψ

$\Rightarrow$ Anwendung auf $|b\rangle$ von QPE: Zustand

$$|0\rangle|b\rangle \mapsto \sum \beta_j |\tilde{\lambda}_j\rangle |u_j\rangle$$

binär-Rep. von λ_j

3. Controlled rotation: füge Ancilla qubit hinzu und drehe um Winkel (Dreh. mit $R_y(2\Theta)$ definiert)

$$\frac{\Theta}{2} = \arcsin\left(\frac{C}{\tilde{\lambda}_j}\right)$$

$$\Rightarrow \sum_j \beta_j |\tilde{\lambda}_j\rangle |u_j\rangle \left(\sqrt{1-\frac{C^2}{\tilde{\lambda}_j^2}}\,|0\rangle + \frac{C}{\tilde{\lambda}_j}|1\rangle\right)$$

hier: C.. Konstante s.d.

$$C \leq \min|\lambda_j| = \Theta\left(\frac{1}{\kappa}\right)$$

<u>Bem.</u>: $\arcsin(\alpha)$ kann mit $\mathcal{O}(\mathrm{poly}(n))$ elementaren gates (approx.) realisiert werden
$\rightarrow$ Literatur

4. Inverse QPE „uncomputing": Zustand

$$\sum_{j=0}^{N-1} \beta_j |0\rangle |u_j\rangle \left(\sqrt{1-\frac{c^2}{\tilde{\lambda}_j^2}}\,|0\rangle + \frac{c}{\tilde{\lambda}_j}|1\rangle\right)$$

5. Messen des letzten Qubits
Falls Ergebnis $|1\rangle \Rightarrow$ Zustand

$$c \sum_{j=0}^{N-1} \frac{\beta_j}{\tilde{\lambda}_j} |0\rangle |u_j\rangle$$

propor. zu $|\tilde{x}\rangle$

Wahrsch. $|1\rangle$ zu messen: $\dfrac{1}{\sqrt{\sum_{j=0}^{N-1} \frac{c^2 |\beta_j|^2}{|\tilde{\lambda}_j|^2}}}$

<u>Bem.</u> Normalisierungskonstante kürzt c
$\Rightarrow$ tritt nicht in Lsg. auf, aber sehr wohl in der Erfolgswahrsch.

Quantum composer Bsp

$$A = \begin{pmatrix} 1 & -\frac{1}{3} \\ -\frac{1}{3} & 1 \end{pmatrix} \qquad b = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$ 1-qubit System

Lsg. von $Ax = b$ $\rightsquigarrow x = \frac{1}{8}\begin{pmatrix} 3 \\ 9 \end{pmatrix}$

EW/EV von A: $\lambda_0 = \frac{2}{3}$, $\lambda_1 = \frac{4}{3}$

$$v_0 = \frac{1}{\sqrt{2}}\begin{pmatrix} -1 \\ -1 \end{pmatrix} \quad v_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

brauche $U = e^{iAt}$ wähle $t = \frac{3\pi}{4}$

$U^2 = e^{i2At}$ $\rightsquigarrow$ exakte Binärdarst. für EWe in QPE mit 2 qubits

$$U = \frac{1}{\sqrt{2}}\begin{pmatrix} -1 & -1 \\ -1 & 1 \end{pmatrix} \underbrace{\begin{pmatrix} e^{i\lambda_0 t} & 0 \\ 0 & e^{i\lambda_1 t} \end{pmatrix}}_{\begin{pmatrix} i & 0 \\ 0 & -1 \end{pmatrix}} \frac{1}{\sqrt{2}}\begin{pmatrix} -1 & -1 \\ -1 & 1 \end{pmatrix}$$

$$= \frac{1}{2}\begin{pmatrix} -1+i & 1+i \\ 1+i & -1+i \end{pmatrix}$$

analog $U^2 = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$

Implementiert mit 4-Parameter unitary

Gate in IBM-Q:

$$U = \begin{pmatrix} e^{i\gamma}\cos(\theta/2) & -e^{i(\gamma+\lambda)}\sin(\theta/2) \\ e^{i(\gamma+\varphi)}\sin(\theta/2) & e^{i(\gamma+\varphi+\lambda)}\cos(\theta/2) \end{pmatrix}$$

$\rightarrow$ $\theta = \frac{\pi}{2}$, $\varphi = -\frac{\pi}{2}$, $\lambda = \frac{\pi}{2}$, $\gamma = \frac{3\pi}{4}$ für U

$\theta = \pi$, $\varphi = \pi$, $\lambda = 0$, $\gamma = 0$ für U^2

Controlled rotation

EWe aus QPE $\tilde{\lambda}_j = N \lambda_j t / 2\pi$ ← hier 2 qubits → 4

$\rightarrow \tilde{\lambda}_0 = 1 \quad \tilde{\lambda}_1 = 2$

$\rightarrow$ kann Konstante $C = 1$ wählen

$\rightarrow \theta_1 = 2\arcsin\left(\frac{1}{\tilde{\lambda}_1}\right) = \pi$

$\theta_2 = 2\arcsin\left(\frac{1}{\tilde{\lambda}_2}\right) = \frac{\pi}{3}$

Resultat: bekomme normierte Wahrscheinlichkeiten

Zust.	P
$\lvert 00\rangle$	0.1875
$\lvert 01\rangle$	0.0625
$\lvert 10\rangle$	0.1875
$\lvert 11\rangle$	0.5625

↑ $\lvert\rangle$-qubit ↖ Ancilla

nur relevant, da Ancilla $\lvert 1\rangle$ gemessen werden muss

cond. Messung ↳ von Ancilla $\lvert 0\rangle$ mit $P = \frac{1}{10}$, $\lvert 1\rangle$ mit $P = \frac{9}{10}$

Wahrsch.: Verhältnis 1:9

norm. Lösung: $\lvert x\rangle = \frac{3}{\sqrt{90}}\lvert 0\rangle + \frac{9}{\sqrt{90}}\lvert 1\rangle$

ebenfalls!

Fehler- und Komplexitätsanalyse

bisherige Ann.: alle Größen haben <u>exakte</u> Binärdarst.
i.A. nicht machbar für EW von A

$\rightsquigarrow$ Phase estimation produziert EWe $\tilde{\lambda}_j$ mit

$$|\tilde{\lambda}_j - \lambda_j| \leq \delta$$

mit Wahrsch. $1 - \frac{1}{poly(n)}$ mit
Laufzeit $\mathcal{O}(T_U \, poly(n) / \delta)$ wobei
T_U – Rechenzeit für Impl. von $U = e^{iA}$

<u>Problem</u>: numerische Stabilität von $\lambda_j \mapsto 1/\lambda_j$
mit contr. Rotation

Falls $\lambda_j \approx 0$ $\rightsquigarrow$ kleine Störung $\tilde{\lambda}_j = \lambda_j + \varepsilon$

$\rightsquigarrow$ großen Effekt auf Fehler:

$$\left| \frac{1}{\lambda_j} - \frac{1}{\tilde{\lambda}_j} \right| = \left| \frac{\varepsilon}{\lambda_j(\lambda_j + \varepsilon)} \right| \approx \frac{\varepsilon}{\lambda_j^2}$$

signif. größer als ε !

$\rightsquigarrow$ Problem mit schlechter Konditionszahl κ von A !

Lösg: Filter Funktionen

Invertiere EWe nur sofern $\lambda \geq \frac{1}{K}$ ← Kond. Zahl

$\rightsquigarrow f(\lambda) = 0$ für $\lambda < \frac{1}{2K}$

für große EW $\sim \frac{1}{\lambda}$

$$\Rightarrow f(\lambda) = \frac{1}{2K\lambda} \quad \lambda \geq \frac{1}{K}$$

dazwischen interpolierend (für num. Stab.)

$$f(\lambda) = \frac{1}{2}\sin\left(\frac{\pi}{2}\cdot(2K\lambda - 1)\right) \quad \frac{1}{2K} \leq \lambda < \frac{1}{K}$$

$\Rightarrow f$ stetig

analog umgek. Filter

$$g(\lambda) = \begin{cases} 0 & \lambda \geq \frac{1}{K} \\ \frac{1}{2}\cos\left(\frac{\pi}{2}(2K\lambda - 1)\right) & \frac{1}{K} > \lambda > \frac{1}{2K} \\ \frac{1}{2} & \frac{1}{2K} \geq \lambda \end{cases}$$

$\Rightarrow g$ stetig und $f^2(\lambda) + g^2(\lambda) \leq 1 \quad \forall \lambda$

Anm: nur eine Wahl, nicht eindeutig,
auch Abschneiden bei $\frac{1}{2K}$ kann woanders sein

statt controlled rotation füge 3-qubit register hinzu

also keine Inversion durchgeführt.

$$|h(\tilde{\lambda}_j)\rangle := \sqrt{1-f(\tilde{\lambda}_j)^2-g(\tilde{\lambda}_j)^2}\,|nothing\rangle + f(\tilde{\lambda}_j)\,|well\rangle + g(\lambda_j)\,|ill\rangle$$

$f(\tilde{\lambda}_j)$: hier EWe invertiert

$g(\lambda_j)$: Teile von $|b\rangle$ im schlecht kond. TR von A

danach $(QPE)^{-1}$ und Messen von $|well\rangle$

<u>prob:</u> Endzustand $\approx \sum_{j:\lambda_j \geq \frac{1}{\kappa}} \lambda_j^{-1} \beta_j |u_j\rangle |well\rangle + \sum_{j:\lambda_j < \frac{1}{\kappa}} \beta_j |u_j\rangle |ill\rangle$

<u>Lemma</u> Die Abb. $\lambda \mapsto |h(\lambda)\rangle$ ist Lipschitz stetig mit $L = \mathcal{O}(\kappa)$, i.e.:

$$\| |h(\lambda_i)\rangle - |h(\lambda_j)\rangle \|_2 \leq C\kappa |\lambda_i - \lambda_j|$$

<u>Bew:</u> Elementar durch Abschätzen der Abl. von f, g (expl. berechenbar)

$\square$

<u>Ziel</u>: Fehlerabsch. für inexakte QPE nach gefilterter Inversion der EW

exakt: $|\psi\rangle := \sum \beta_i |u_i\rangle |h(\lambda_i)\rangle$

approx.: $|\tilde{\psi}\rangle := \sum \beta_i |u_i\rangle |h(\tilde{\lambda}_i)\rangle$

$$\Rightarrow \| |\psi\rangle - |\tilde{\psi}\rangle \|_2^2 = \| |\psi\rangle \|^2 + \| |\tilde{\psi}\rangle \|^2 - 2\,\mathrm{Re}\langle\psi|\tilde{\psi}\rangle$$

$$= 2\big(1 - \underbrace{\mathrm{Re}\langle\psi|\tilde{\psi}\rangle}_{\in[0,1]\ \text{(C.S.)}}\big)$$

$$\Rightarrow \mathrm{Re}\langle\psi|\tilde{\psi}\rangle \underset{u_i\ \text{ONB}}{=} \sum_{i=1}^{N} |\beta_i|^2\, \mathrm{Re}\langle h(\lambda_i)|h(\tilde{\lambda}_i)\rangle$$

$$\text{Lemma} \Rightarrow \mathrm{Re}\langle h(\lambda_i)|h(\tilde{\lambda}_i)\rangle \geq 1 - \frac{C^2K^2}{2}|\lambda_i - \tilde{\lambda}_i|^2$$

$$\underset{\text{Fehler pro EW} \leq \delta}{\geq} 1 - \frac{C^2K^2\delta^2}{2}$$

$$\leq \underbrace{\sum_{i=1}^{N} |\beta_i|^2}_{=1} C^2K^2\delta^2$$

$$\Rightarrow \| |\psi\rangle - |\tilde{\psi}\rangle \|_2 \leq CK\delta$$

Für Fehler $\mathcal{O}(\varepsilon)$

↳ Phase estimation Fehler $\delta = \mathcal{O}\left(\frac{\varepsilon}{\kappa}\right)$

↳ Runtime $\mathcal{O}(\kappa \, \mathrm{poly}(n)/\varepsilon)$

Messung: möchte Ancilla $|1\rangle$ (wohlkond. A)

Ancilla $|\mathrm{well}\rangle$ (generelleres A)

⇝ Erfolgswahrsch

$$p \geq \sum_{i:\lambda_i \geq 1/\kappa} |\beta_i|^2 \left|\frac{1}{\lambda_i \kappa}\right|^2 = \mathcal{O}\left(\frac{1}{\kappa^2}\right)$$

↑ note: $\sum \frac{|\beta_i|^2}{|\lambda_i|^2} = \|A^{-1}b\|$

kann mittels „amplitude amplification" (vgl. Grover-Algorithmus) auf $\mathcal{O}\left(\frac{1}{\kappa}\right)$ verbessert werden

⇒ $\mathcal{O}(\kappa)$ von Prozedur nötig um $|\mathrm{well}\rangle$ mit bel. hoher Wahrsch. zu erhalten

Gesamtaufwand:

·) state prep.: $\mathcal{O}(n)$ (falls QRAM precomputed)

·) Hamiltonian simulation:

e^{iAt} wenn A s-sparse bis auf Fehler ε in

$$\mathcal{O}\left(n s t \, \mathrm{poly}\left(\log \frac{st}{\varepsilon}\right)\right)$$

(bestes Resultat in Literatur, einfache Methoden in VO vorgestellt $\mathcal{O}(n s^2 t^2/\varepsilon)$)

·) QPE $\mathcal{O}(\mathrm{poly}(n)\, K/\varepsilon \cdot T_U)$

·) Ampl. amp. $\mathcal{O}(K)$

$$\Rightarrow \mathcal{O}\left(\mathrm{poly}(n)\, K^2 s/\varepsilon \cdot \mathrm{poly}\left(\log \frac{s}{\varepsilon}\right)\right)$$

vgl. CG:

$$\mathcal{O}\left(\exp(n)\, s\sqrt{K}\, \ln\left(\frac{2}{\varepsilon}\right)\right)$$

besser in n aber schlechter in K, ε

exp. speedup!

Verbesserbar??

Verbesserung des HHL-Algorithmus

CG: Komplexität vs. Genauigkeit $\mathcal{O}(\log \frac{1}{\varepsilon})$

HHL: wegen QPE $\longrightarrow$ $\mathcal{O}(\frac{1}{\varepsilon})$

Ziel: q-LGS Algorithmus mit ebenfalls log. Abh. von ε^{-1}, immer noch exp. speedup in N

Idee: Approximiere A^{-1} direkt

vgl. Cayley-Hamilton Theorem:

p.-char. Polynom von A $\Rightarrow p(A)=0$

$\in \mathcal{P}_N$ $\quad \Rightarrow A^N + \alpha_{N-1} A^{N-1} + \ldots + \alpha_0 I = 0 \qquad \alpha_i \in \mathbb{C}$

$$\Leftrightarrow A^{-1} = -\frac{1}{\alpha_0} \cdot \left(\alpha_1 I + \ldots + \alpha_{N-1} A^{N-2} + A^{N-1}\right)$$

$$= p_{N-1}(A)$$

Berechnung von α_i zu teuer, aber

ev. $\exists$ Polynom $q_m \in \mathcal{P}_m$ mit $m \ll N-1$

s.d. $A^{-1} \approx q_m(A)$

und $q_m(A)$ ist eff. implementierbar

2 Mögl. 1. trigonometrische Polynome $\rightsquigarrow$ Fourier Approx.
$\rightarrow$ unitär

2. Chebyshev Polynome $\quad \|q_m(A)\|_2$
$\rightsquigarrow$ minimieren

Q: Wie wirkt sich approx. der Matrix auf Endzustand aus?

Lemma Sei B hermitisch mit $\|B^{-1}\| \leq 1$ und D so dass $\|B-D\| \leq \varepsilon < \frac{1}{2}$

$\Rightarrow$ Zustände $|x\rangle := \frac{B|\psi\rangle}{\|B|\psi\rangle\|}$ und $|\tilde{x}\rangle := \frac{D|\psi\rangle}{\|D|\psi\rangle\|}$ erfüllen $\| |x\rangle - |\tilde{x}\rangle \| \leq 4\varepsilon$

Beweis: Δ-Ungleichg.:

$$\| |x\rangle - |\tilde{x}\rangle \| = \left\| \frac{B|\psi\rangle}{\|B|\psi\rangle\|} - \frac{D|\psi\rangle}{\|D|\psi\rangle\|} \right\|$$

$$\leq \underbrace{\left\| \frac{B|\psi\rangle}{\|B|\psi\rangle\|} - \frac{B|\psi\rangle}{\|D|\psi\rangle\|} \right\|}_{\leq \frac{\left| \|D|\psi\rangle\| - \|B|\psi\rangle\| \right|}{\|D|\psi\rangle\|}} + \frac{1}{\|D|\psi\rangle\|} \| B|\psi\rangle - D|\psi\rangle \|$$

nochmal Δ-Ungleichg.:

$$1 \underset{\text{bel. all.}}{\leq} \| B|\psi\rangle \| \leq \| D|\psi\rangle \| + \| (B-D)|\psi\rangle \| \leq \| D|\psi\rangle \| + \varepsilon$$

$\Rightarrow$ beide Terme $\leq \frac{\varepsilon}{\|D|\psi\rangle\|} + \frac{\varepsilon}{\|D|\psi\rangle\|} \leq 2 \frac{\varepsilon}{1-\varepsilon} \leq 4\varepsilon$ $\square$

$\Rightarrow$ wende Lemma mit $B = A^{-1}$ und $D \approx A^{-1}$ an

---

<u>Fourier approach</u>

Approximiere $A^{-1} \approx \sum_j \alpha_j e^{-iAt_j}$ $\quad \alpha_j, t_j \in \mathbb{R}$

also durch <u>Linearkomb. von unitären Op.</u> (LCU)

<u>2 Fragen</u>: ·) Wie genau sieht die Approx. aus?

·) Ist diese effizient implementierbar?

<u>Implementierung von LCU</u>

oBdA $\alpha_j > 0$ da Phase sowieso nicht messbar

<u>Ziel</u>: Impl. von $M = \sum_j \alpha_j U_j$, wobei U_j unitär

M nicht unbedingt!

<u>Lemma</u> Sei $M = \sum_j \alpha_j U_j$ mit $\alpha_j > 0$, U_j unitär

und V Abb. $V|0^m\rangle := \frac{1}{\sqrt{\alpha}} \sum_j \sqrt{\alpha_j}\, |j\rangle$ mit $\alpha = \sum_j \alpha_j$

sowie $U := \sum_j |j\rangle\langle j| \otimes U_j$

$\Rightarrow$ $V^H U V =: W$ erfüllt

$$W|0^m\rangle|\psi\rangle = \frac{1}{\alpha}|0^m\rangle M|\psi\rangle + |\phi^\perp\rangle$$

$\forall$ zust. $|\psi\rangle$, wobei $(|0^m\rangle\langle 0^m| \otimes I)|\phi^\perp\rangle = 0$
$=: \Pi|\phi^\perp\rangle$

Also: 1. Term realisiert M (nachher A^{-1})

2. Term orthogonal zu $|0^m\rangle$ im 1. Register

$\Rightarrow$ Messen von $|0^m\rangle$ im 1. Reg. $\rightsquigarrow$ Impl. von M

<u>Beweis</u>: $W|0^m\rangle|\psi\rangle = V^H U\left(\frac{1}{\sqrt{\alpha}}\sum_j \sqrt{\alpha_j}|j\rangle|\psi\rangle\right)$

$$\overset{\text{select } U_j}{=} V^H\left(\frac{1}{\sqrt{\alpha}}\sum_j \sqrt{\alpha_j}\,|j\rangle U_j|\psi\rangle\right)$$

$$= \underbrace{\Pi V^H(\quad)}_{} + \underbrace{(I-\Pi)V^H(\quad)}_{}$$

$(|0^m\rangle\langle 0^m|\otimes I)\left(\frac{1}{\sqrt{\alpha}}\sum\sqrt{\alpha_j}\langle j|\otimes I\right)|\phi^\perp\rangle$ da $\Pi(I-\Pi)=0$

da Π 1. Reg. auf $|0^m\rangle$ Proj.
und aus Def. von V^H

$\Rightarrow$

$$= \frac{1}{\alpha}|0^m\rangle\sum_j \alpha_j U_j|\psi\rangle + |\phi^\perp\rangle$$

$\square$

·) U -- „select U", wählt U_i anhand von Kontrollregister aus

·) Erfolgswahrsch. $\frac{\| M|\psi\rangle \|^2}{\alpha^2}$

unsere Anwendung $\rightarrow M = A^{-1}$, $|\psi\rangle = |b\rangle$

mit QRAM: multiple $|b\rangle$-Preparation möglich

↳ Amplitude amplification möglich (Drehung um $|b\rangle$) $\Rightarrow$ quadratischer Speedup

hohe Wahrsch. in $\mathcal{O}\left(\frac{\alpha}{\| M|\psi\rangle \|}\right)$ Wiederholungen

·) V -- unitär, einach impl.

·) falls unitäre Op. U_i einfach implementierbar und Zerlegung $\mathcal{O}(\log N)$-Terme

↳ eff. Impl. von M

·) Query-Komplexität von $U \simeq$ Query-Kompl. des teuersten U_j

für Gate-Komplexität i. A. nicht so gut!

<u>aber</u>: für spez. Anw. hier schon, da $U_j = (e^{-iA})^{t_j}$ überall gleich

Diagonalisierung $\quad A = T^H D T$

$$\Rightarrow \quad T^H D^{-1} T = A^{-1} \overset{!}{\approx} \sum_j \alpha_j e^{-iAt_j}$$

$$= T^H \sum_j \alpha_j e^{-iDt_j} T$$

da $D = \mathrm{diag}(\lambda_j) \Rightarrow$ brauche nur Entwicklung von

$$f(x) = \frac{1}{x} \overset{!}{\approx} \sum_j \alpha_j e^{-ixt_j}$$

für $x \in \sigma(A)$ bzw. $x \in \Omega \supseteq \sigma(A)$

---

wie bei HHL $\rightsquigarrow$ betrachte nur gut konditionierte EWe ($\geq \frac{1}{\kappa}$), verlange auch $\lambda_{max} = 1$ (skalierung)

$\rightsquigarrow$ suche Approx. an f auf $D_\kappa := [-1,1] \setminus [-\frac{1}{\kappa}, \frac{1}{\kappa}]$

<u>Idee</u>: 1) glätte f um 0

2) Fouriertrafo (Integral)

3) Ersetze Integral durch endl. Summe

$\rightarrow$ Funktion h der Form $\sum \alpha_j e^{-ixt_j}$ mit

$$\sup_{x \in D_\kappa} |f(x) - h(x)| \leq C\varepsilon$$

$\rightarrow$ Vorherige Lemmata $\rightsquigarrow$ $h(A)|b\rangle$ (+Normierung) ist Approx. an $|x\rangle$

Lemma Die Funktion

$$h(x) := \frac{i}{\sqrt{2\pi}} \sum_{j=0}^{J-1} \sum_{k=-K}^{K} \Delta_y \Delta_z z_k e^{-\frac{z_k^2}{2}} e^{-ixy_j z_k}$$

mit $y_j := j\Delta_y$, $z_k := k\Delta_z$ und $J = \mathcal{O}\left(\frac{K}{\varepsilon} \log \frac{K}{\varepsilon}\right)$,
$K = \mathcal{O}\left(K \log \frac{K}{\varepsilon}\right)$, $\Delta_y = \mathcal{O}\left(\frac{\varepsilon}{\sqrt{\log K/\varepsilon}}\right)$, $\Delta_z = \mathcal{O}\left(\frac{1}{K\sqrt{\log \frac{K}{\varepsilon}}}\right)$
erfüllt

$$\sup_{x \in D_K} \left| h(x) - \frac{1}{x} \right| \leq c\varepsilon$$

Bew: Sei $g(y) = y e^{-\frac{y^2}{2}}$

Subst. $\Rightarrow \frac{1}{x} = \int_0^\infty g(xy)\,dy$ da $\int_0^\infty g(y)\,dy = 1$

Wahl von $g \Rightarrow$ klingt schnell ab, glatt und

$\mathcal{F}(g) = -i g$ also Eigenfkt. von Fourier trafo zu EW $-i$

$$\Rightarrow g = i\mathcal{F}(g) = \frac{i}{\sqrt{2\pi}} \int_{\mathbb{R}} z e^{-\frac{z^2}{2}} e^{-iyz}\,dz$$

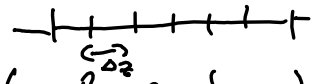
$$\Rightarrow \frac{1}{x} = \frac{i}{\sqrt{2\pi}} \int_0^\infty \int_{-\infty}^\infty z e^{-\frac{z^2}{2}} e^{-ixyz}\,dz\,dy$$

$\rightsquigarrow$ $h(x)$ ist Riemann Summe zu abgeschnittenen Integralen in Formel für $\frac{1}{x}$

<u>1. Schritt</u>: Summation / Integr. in y: in h $\rightsquigarrow$ geom. Reihe

$$h(x) = \frac{i\Delta_y}{\sqrt{2\pi}} \sum_{k=-K}^{K} \Delta_z \, z_k e^{-\frac{z_k^2}{2}} \frac{1-e^{-ixy_j z_k}}{1-e^{-ix\Delta_y z_k}}$$

$$\frac{1}{x} = \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{x} \int_{-\infty}^{\infty} e^{-\frac{z^2}{2}} \, dz$$



<u>2. Schritt</u>: Approx von $\uparrow$ durch Reihe (∞-Riemann Summe)

$$\left| \frac{1}{x} \cdot \Big(1 - \underbrace{\frac{1}{\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} \overset{\text{Gitterweite}}{\Delta_z} e^{-\frac{z_k^2}{2}}}_{= \sum_{k=-\infty}^{\infty} e^{-\left(\frac{2\pi k}{\Delta_z}\right)^2/2} = 1 + \sum_{|k|\geq 1} e^{-\left(\frac{2\pi k}{\Delta_z}\right)^2/2}} \Big) \right|$$

Poisson'sche Summenformel

$$\sum_{k=-\infty}^{\infty} f(k) = \sqrt{2\pi} \sum_{k=-\infty}^{\infty} \hat{f}(2\pi k)$$

$$\Rightarrow \leq \frac{1}{|x|} \cdot 2 \sum_{k=1}^{\infty} e^{-\frac{2\pi^2 k}{\Delta_z^2}} \overset{\text{geom. Reihe}}{=} \frac{1}{|x|} \frac{2}{e^{2\pi^2/\Delta_z^2} - 1}$$

$$\leq C K \cdot \varepsilon e^{-K^2 \log K}$$ nach Wahl von Δ_z und mit $\frac{1}{|x|} \leq K$

$$\leq \tilde{C} \varepsilon$$

3. Schritt: Reihe abschneiden

$$\Rightarrow \left| \frac{1}{\sqrt{2\pi}x} \cdot \left(\sum_{k=-\infty}^{\infty} \Delta z \cdot e^{-\frac{z_k^2}{2}} - \sum_{k=-K}^{K} \Delta z \cdot e^{-\frac{z_k^2}{2}} \left(1 - e^{-ixy_j z_k}\right) \right) \right|$$

$$\leq C\varepsilon$$

folgt aus Dreiecksungl. (füge Term $(1-e^{-ixy_j z_k})$ hinzu)

Poisson'sche Summenformel

Abschätzung Reihenrest mittels Wahl von K

Länglich, aber einfach $\rightsquigarrow$ Literatur

4. Schritt:

$$\left| h(x) - \frac{1}{\sqrt{2\pi}x} \sum_{k=-K}^{K} \Delta z \, e^{-\frac{z_k^2}{2}} \left(1 - e^{-ixy_j z_k}\right) \right|$$

Formel für h aus 1. Schritt:

$$\leq \frac{2}{\sqrt{2\pi}} \left| \sum_{k=-K}^{K} \underbrace{\left(\frac{i \Delta y z_k}{1 - e^{-ix\Delta y z_k}} - \frac{1}{x} \right)}_{\leq \Delta y |z_k| \text{ da } \left| \frac{1}{1-e^{-ix}} - \frac{1}{x} \right| < 1 \text{ für } x \in [-1,1]} \cdot \Delta z \, e^{-\frac{z_k^2}{2}} \right|$$

$$\overset{|z_k| \leq e^{\frac{|z_k|^2}{4}}}{\leq} \sqrt{\frac{2}{\pi}} \, \Delta y \underbrace{\sum_{k=-K}^{K} \Delta z \, e^{-\frac{z_k^2}{4}}}_{\leq \int_0^\infty e^{-\frac{z^2}{4}} dz}$$

$$= C \Delta y \leq C\varepsilon \qquad \text{up to lag-terms}$$

<u>5. Schritt</u>: Kombiniere Schritt 2–4 mit Dreiecksungl.

$$\Rightarrow \left| h(x) - \frac{1}{x} \right| \leq C \cdot \varepsilon \qquad \square$$

<u>Komplexität</u>

<u>Theorem</u> Das q-LGS kann mit $\mathcal{O}(K\sqrt{\log \frac{K}{\varepsilon}})$ Anwendungen von Hamiltonian Sim. für e^{-iAt} mit $t = \mathcal{O}(K \log \frac{K}{\varepsilon})$ mit Genauigkeit $\mathcal{O}\left(\frac{\varepsilon}{K\sqrt{\log \frac{K}{\varepsilon}}}\right)$ gelöst werden.

Die Gate-Komplexität ist

$$\mathcal{O}\left(sK^2 \log^{2.5}\left(\frac{K}{\varepsilon}\right)\left(\log N + \log^{2.5}\left(\frac{K}{\varepsilon}\right)\right)\right)$$

<u>Beweis</u> $\rightsquigarrow$ Literatur muss V, U impl.

V in $\mathcal{O}(K \log \frac{K}{\varepsilon})$

U bzw. select $U_{\sim}$ mit Ham. Simulation

für sparse $\rightsquigarrow \mathcal{O}\left(st\left(\log N + \log^{2.5}\left(\frac{t}{\varepsilon}\right)\right)\log \frac{t}{\varepsilon}\right)$

Wahl von $t \rightsquigarrow$ gew. Genauigkeit

Ham. Sim. muss $\mathcal{O}(\underbrace{K\sqrt{\log \frac{K}{\varepsilon}}}_{\alpha \text{ in Lemma vorne}})$ durchgeführt werden $\square$

Chebyshev approach

Idee: ersetze Fourier-Entwicklung durch Chebyshev Polynome

Problem: sind nicht unitär

Lösg. „Block-encoding" (n+1-qubit block enc.)

Sei $A \in \mathbb{C}^{2^n \times 2^n}$, $\|A\| \leq 1$ und angenommen, dass $\exists$ unitäre Matrix $U \in \mathbb{C}^{2^{n+1} \times 2^{n+1}}$ s.d.

$$U = \begin{pmatrix} A & * \\ * & * \end{pmatrix}$$

hier $* \in \mathbb{C}^{2^n \times 2^n}$

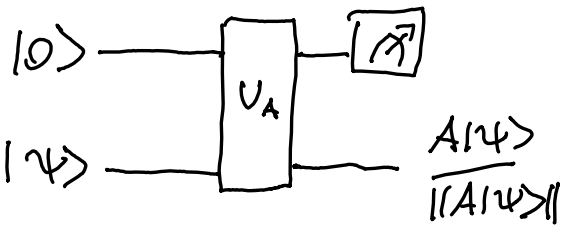
Einträge egal, nur wichtig, dass U unitär

$\rightarrow$ $\forall$ Zustände $|\psi\rangle \rightsquigarrow$ betrachte $|0\rangle|\psi\rangle \simeq \begin{pmatrix} \psi \\ 0 \end{pmatrix}$

$$\Rightarrow U(|0\rangle|\psi\rangle) = |0\rangle A|\psi\rangle + \underbrace{|1\rangle|\phi^\perp\rangle}_{\text{unwichtig}}$$

$\Rightarrow$ messe $|0\rangle$ in 1. qubit $\rightarrow$ Realisierung von $A|\psi\rangle$

circuit:



Bem. ·) LCU (ohne amplitude amp.) ist Spezialfall von block encoding

·) für generelles A → teuer
für sparse A → effizient durchführbar

⇒ Ziel: block encoding von A^{-1}

Theorem Sei $p \in \mathcal{P}_d$ mit $\|p\|_{\infty,[-1,1]} \leq \frac{1}{4}$ polynom von Grad $\leq d$. Sei U block enc. von A mit $(n+a)$-qubits.
⇒ Ein block encoding von $P(A)$ mit $(n+\mathcal{O}(a))$-qubits kann mittels d-Anwendungen von U, U^{-1}, einer Anwendung von controlled-U und $\mathcal{O}(ad)$ elem. Gates (2-qubit) realisiert werden.

Anwendung von Thm.:

Approx. $f(x) = \frac{1}{x}$ durch $p(x)$:

für $\tilde{d} = \mathcal{O}(K^2 \log \frac{K}{\varepsilon})$

$$\Rightarrow \sup_{x \in D_K} \Big| \underbrace{\frac{1-(1-x^2)^{\tilde{d}}}{x}}_{\in \mathcal{P}_{2\tilde{d}-1}} - \frac{1}{x} \Big| \leq \frac{\varepsilon}{2}$$

·) schreibe $\frac{1-(1-x^2)^{\tilde{d}}}{x} = \sum_{j=0}^{2\tilde{d}-1} \alpha_j T_j$

↳ Cheby. polynome

·) schneide Entwicklung bei $d = \mathcal{O}(K \log \frac{K}{\varepsilon})$ ab

→ Fehler $\leq \frac{\varepsilon}{2}$

⇒ Theorem gibt eff. Implementierung von $P_d(A)$

⇒ Anwendung von block encoding → Zustand $|\tilde{x}\rangle$ mit

$$\| |\tilde{x}\rangle - |x\rangle \| \leq C\varepsilon$$

<u>gesamt</u>: Gate-komplexität:

$$\mathcal{O}\left(s K^2 \log^2\left(\frac{sK}{\varepsilon}\right)\left(\log N + \log^{2.5}\left(\frac{sK}{\varepsilon}\right)\right)\right)$$

besser als Fourier approach, braucht aber direkt sparse access (funktioniert nicht für allg. Matrizen)

Fourier approach → statt dessen Hamiltonian simulation

↳ allgemeiner