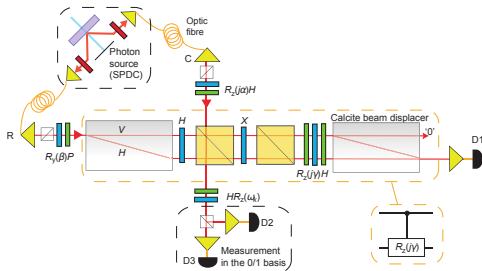


Simulation quantenmechanischer Systeme

Hauptseminar: *Physik des Quantencomputers (SS13)*

Johannes Janssen

Institut für Theoretische Festkörperphysik (TFP)



- Motivation und Ziel
- Beliebige Systeme
- Systeme mit lokaler Wechselwirkung
- Digitale Quantensimulation
- Iterative Phase Estimation Algorithm
- 1. und 2. Quantisierung
- Energien des H_2 -Moleküls

Warum überhaupt simulieren?

- Modell überprüfen, experimentell nicht realisierbar
- keine analytische Lösung

Wie simulieren?

- Wir schauen uns die Zeitentwicklung an, um etwas über das System zu lernen.

Was ist eine gute Simulation?

- skaliert, also Speicher- und Zeitaufwand steigen höchstens *polynomiell* mit
 - Systemgröße
 - Simulationsdauer
 - Genauigkeit

R. P. Feynman (1982): *Simulating Physics with Computers*



- „nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical“
- „spin waves in a spin lattice imitat[e] Bose-particles in the field theory“ → *analoge Quantensimulation*
- Aufgabe: „work out the classes of different kinds of quantum mechanical systems which are really intersimulatable“

beliebiges System

- Systemgröße: N Quantenzahlen
→ Hilbertraum \mathcal{H} wächst exponentiell mit N
- Zeitentwicklung: $U = e^{iHt} \in \mathcal{H}$

N Spin- $\frac{1}{2}$ -Teilchen:

- $\dim \mathcal{H} = 2^N$
- $U \in \mathbb{C}^{2^N \times 2^N}$

klassischer Digitalcomputer

- allein die Wellenfunktion benötigt exponentiell viel Speicher ($\mathcal{O}(2^N)$)
- Zeitentwicklung $\hat{=}$ exponentiell große Matrixgleichung ($\mathcal{O}(4^N)$)

Quantencomputer

- Wellenfunktion benötigt nur polynomiell viele Qbits (N)
- *aber*, da H und damit U beliebig
→ i. A. exponentiell viele Gates nötig um U darzustellen ($\mathcal{O}(4^N)$)

I. A. benötigt auch ein Quantencomputer exponentiell viele Rechenschritte um ein System zu simulieren.

Sei $H = \sum_{i=1}^l H_i$, wobei

- H_i auf einen m_i -dimensionalen Unterraum wirkt,
- mit maximal k der N Quantenzahlen.

Zeitentwicklung

- Suzuki-Trotter: $e^{\delta(A+B)} = e^{\delta A} e^{\delta B} + \mathcal{O}(\delta^2)$

$$\Rightarrow e^{iHt} \approx (e^{iH_1 t/n} \cdot \dots \cdot e^{iH_l t/n})^n \quad \varepsilon \leq \|n(e^{iHt/n} - 1 - iHt/n)\|_{\text{sup}}$$

- nötige Rechenschritte: $T \approx n \sum_{i=1}^l m_i^2 \leq nlm^2$

- normalerweise wächst l nur polynomiell (Modell nächster Nachbarn)

Um Systeme mit lokaler Wechselwirkung zu simulieren, benötigt ein Quantencomputer maximal polynomiell viele Rechenschritte.

Beispiel: Heisenberg-Modell mit N Spin- $\frac{1}{2}$ -Teilchen

$$H = -J \sum_{\langle i,j \rangle} \mathbf{S}_i \mathbf{S}_j$$

- Nachbarn pro Teilchen: $p = 2d$
- H_i beschreibt die Wechselwirkung zwischen zwei Teilchen ($m = 2$)

$$\Rightarrow H = \sum_{i=1}^{p \frac{N}{2}} H_i, \quad H_i = -J \mathbf{S}_{\alpha_i} \mathbf{S}_{\beta_i}$$

- Darstellung der Wellenfunktion N Qbits, nötige Rechenschritte:

$$T \approx n \sum_{i=1}^l m_i^2 = 2nNp \approx 2Np/\varepsilon$$

Vergleich klassischer Digitalcomputer

- Zeitentwicklung kann analog vereinfacht werden
- *aber* allein das Darstellen der Wellenfunktion benötigt exponentiell viel Speicher
- für große Systeme sind daher Näherungslösungen nötig, indem z. B. die erlaubten Wellenfunktionen eingeschränkt werden
 - DFT
 - Monte-Carlo-Simulation

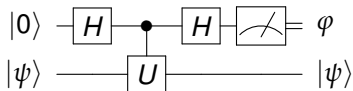
Allgemeine Vorgehensweise

- 1 Zustand präparieren $|0\rangle \rightarrow |\psi\rangle$
 - Grundzustand (z. B. adiabatisch)
 - kanonisches Ensemble (z. B. Metropolis–Algorithmus)
- 2 Zeitentwicklung $|\psi\rangle \rightarrow e^{iHt}|\psi\rangle = e^{2\pi i\varphi}|\psi\rangle$, wobei $H|\psi\rangle = \epsilon|\psi\rangle$, $\epsilon = \frac{2\pi\varphi}{t}$
 - Suzuki–Trotter–Formeln (auch höherer Ordnung)
- 3 Messung von Observablen φ bestimmen
 - Phase Estimation Algorithm (PEA)

(Iterative) Phase Estimation Algorithm

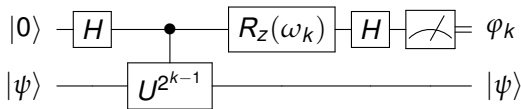
PEA

- schon bekannt vom Shor-Algorithmus



- muss $N \sim 2^{2m}$ durchgeführt werden, um φ auf m binäre Stellen zu bestimmen

IPEA → an der Tafel (PDF)



- m Iterationen, um φ auf m binäre Stellen zu bestimmen

Erste und zweite Quantisierung

am Beispiel eines Moleküls

Erste Quantisierung

- Hamiltonoperator ohne Spinfreiheitsgrade:

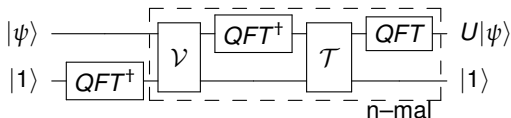
$$H = \underbrace{-\sum_i \frac{\hbar^2}{2M_i} \nabla_i^2 - \sum_j \frac{\hbar^2}{2m_e} \nabla_j^2}_T + \underbrace{\frac{e^2}{4\pi\epsilon_0} \left(\sum_{i<j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} + \sum_{i<j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i,j} \frac{Z_i}{|\mathbf{R}_i - \mathbf{r}_j|} \right)}_V$$

- Diskretisierung von kontinuierlichen Variablen

- Zeitentwicklung:

$$e^{iHt} \approx (U_{\text{QFT}}^\dagger e^{iTt/n} U_{\text{QFT}} e^{iVt/n})^n$$

		0		1					
		00		01		10		11	
000	001	010	011	100	101	110	111		
x= 0	1	2	3	4	5	6	7		



Erste und zweite Quantisierung

am Beispiel eines Moleküls

Zweite Quantisierung

- semi-klassisch (Born–Oppenheimer–Näherung), d.h. Kernpositionen \mathbf{R}_i sind klassische Parameter der Simulation

- Übergang in den Fock–Raum:

$$H \rightarrow H(a_j^\dagger, a_j) = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s,$$

dabei sind h_{pq} und h_{pqrs} klassische Integrale

- Jordan–Wigner–Transformation:

$$a_j^\dagger = \left(\prod_{m<j} \sigma_m^z \right) \sigma_j^-, \quad a_j = \left(\prod_{m<j} \sigma_m^z \right) \sigma_j^+$$

- auf Basis von Photonen realisiertes Experiment
- atomare $|1s\rangle$ -Orbitale bilden (anti-)bindendes Molekülorbital: $|g\rangle, |u\rangle$
- vier Einzelelektronen-Zustände: $|g \uparrow\rangle, |g \downarrow\rangle, |u \uparrow\rangle, |u \downarrow\rangle$
- antisymmetrische Kombination mittels Slater-Determinate:

$$|\Phi_1\rangle = |g \uparrow, g \downarrow\rangle = \frac{1}{\sqrt{2}} (|g \uparrow, g \downarrow\rangle - |g \downarrow, g \uparrow\rangle)$$

$$|\Phi_2\rangle = |g \uparrow, u \uparrow\rangle$$

$$|\Phi_3\rangle = |g \uparrow, u \downarrow\rangle$$

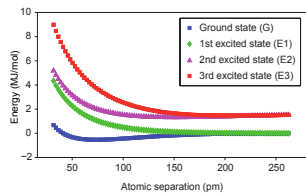
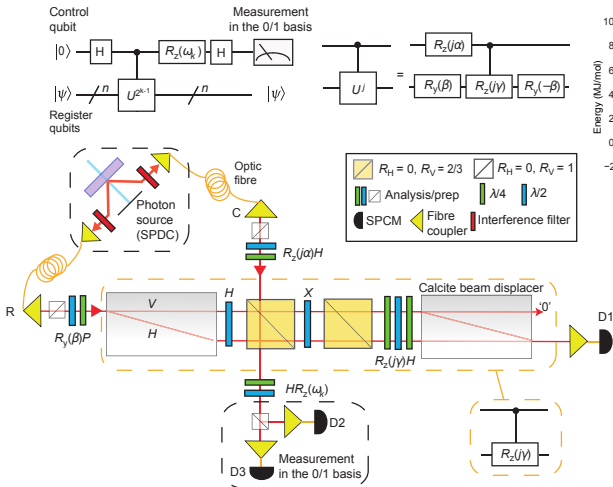
$$|\Phi_4\rangle = |g \downarrow, u \uparrow\rangle$$

$$|\Phi_5\rangle = |g \downarrow, u \downarrow\rangle$$

$$|\Phi_6\rangle = |u \uparrow, u \downarrow\rangle$$

- aus Symmetriegründen ist H blockdiagonal in vier Unterräumen $\{|\Phi_1, \Phi_6\rangle\}, \{|\Phi_2\rangle\}, \{|\Phi_3, \Phi_4\rangle\}, \{|\Phi_5\rangle\}$
- Bestimmung der Eigenwerte von $H^{(1,6)}, H^{(3,4)}$ mittels IPEA

Energien des H_2 -Moleküls



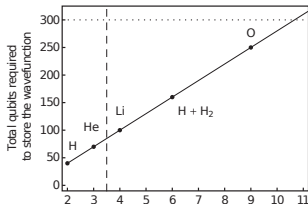
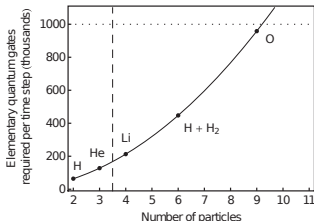
Bildquelle: [Aspuru10]

Nächste Schritte

- Zustand nicht-klassisch präparieren (Grundzustand z.B. adiabatisch)
- direkte Berechnung des Zeitentwicklungsoperators U

Skalierung

- Quantenfehlerkorrektur nötig
- U^{2^k} braucht i. A. doppelt so viele Gates wie U^k
- aber im IEPA wächst die Genauigkeit von 2^k auf 2^{2^k}



Bildquelle: [Aspuru08]

Literatur I



R. P. Feynman

Simulating Physics with Computers

doi:10.1007/BF02650179, 1982



S. Lloyd

Universal Quantum Simulators

doi:10.1126/science.273.5278.1073, 1996



M.–H. Yung, J. D. Whitfield, S. Boixo, D. G. Tempel,
A. Aspuru–Guzik

Introduction to Quantum Algorithms for Physics and Chemistry

arXiv:1203.1331, 2012



M. Dobšíček, G. Johansson, V. Shumeiko, G. Wendin

*Arbitrary accuracy iterative quantum phase estimation algorithm
using a single ancillary qubit: A two-qubit benchmark*

doi:10.1103/PhysRevA.76.030306, 2007



B. P. Lanyon, J. D. Whitfield, G. G. Gillett, M. E. Goggin,
M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell,
M. Barbieri, A. Aspuru-Guzik, A. G. White

Towards quantum chemistry on a quantum computer

doi:10.1038/NCHEM.483, 2010



I. Kassala, S. P. Jordanb, P. J. Lovec, M. Mohsenia, and
A. Aspuru-Guzika

*Polynomial-time quantum algorithm for the simulation of chemical
dynamics*

doi:10.1073/pnas.0808245105, 2008