

Übungen zur Einführung in die Quanteninformationsverarbeitung – Blatt 07 Sommersemester 2018 ¹

Abgabetermin Montag, 18. Juni, vor Vorlesung

VERTIEFUNG DES GROVER-ALGORITHMUS

Um Sie die Funktionsweise des Grover-Algorithmus noch etwas klarer zu machen, werden Sie ihn auf dem Rechner simulieren. Schreiben Sie dazu ein Programm, das für eine Datenbank mit $N = 2^n$ (n möglichst beliebig) Einträgen den Grover-Algorithmus simuliert. Es gebe M Lösungen des Orakels, d.h. genau für alle $|x\rangle \in \mathbf{M}$ mit $M = |\mathbf{M}|$ gilt $\hat{O}|x\rangle = -|x\rangle$, sonst $\hat{O}|y\rangle = |y\rangle$.

Ihre Lösung sollte beispielhaft die folgenden Ergebnisse dokumentieren:

- Die Matrix des Orakeloperators \hat{O} für $n = 3$ und irgendwelche $M = 2$ Lösungen (1 Punkt).
- Die Matrix des Groveroperators für $n = 3$ und irgendwelche $M = 2$ Lösungen (1 Punkt).
- Das Ergebnis $|\psi_k\rangle$ nach der idealen Anzahl von k Schritten (wieder für $n = 3$ und $M = 2$). Was ist die ideale Anzahl von Schritten? (3 Punkte)
- Das Gleiche wie in (c) für $n = 3$ und $M = 1$ (2 Punkte).
- Zeichnen Sie für $n = 7$ und $M = 1$ die Wahrscheinlichkeit, eine Lösung zu messen, gegen die Anzahl der Iterationen (3 Punkte).

Hilfestellung: Die Simulation kann natürlich in einer beliebigen Sprache so wie Mathematica, Maple, Matlab, C etc. durchgeführt werden. Hier noch ein paar Tips zum Verwenden von Mathematica:

- Mit der Hilfsfunktion F1 kann man jederzeit auf das komplette Mathematica-Handbuch zurückgreifen. Die Befehle sind alle so mit Beispielen dokumentiert, dass man sich schnell einlernt.
- Listen, Vektoren und Matrizen kann man gut mit der Funktion `Table` erzeugen, so ergibt z.B. `A=Table[If[m+n==2 n, 1, 0], {m, 1, 5}, {n, 1, 5}]` eine 5x5 Einheitsmatrix, die sich schön als Matrix mit dem Befehl `A//MatrixForm` darstellen lässt.
- Für die Orakel-Funktion kann es praktisch sein, den Befehl `MemberQ[Liste, value]` zu benutzen, der überprüft, ob `value` in der `Liste` enthalten ist.
- Datenpunkte [wie für Aufgabe (e)] benötigt, lassen sich z.B. mit `ListPlot[Liste]` darstellen.
- Die Definition von Funktionen ist hier an einem Beispiel gezeigt: Wenn man `MyUnit[mfinal_, nfinal_, diagonal_] := Table[If[m+n==2n, diagonal, 0], {m, 1, mfinal}, {n, 1, nfinal}]` definiert hat, erzeugt `A=MyUnit[4, 4, 3]` eine 4x4 Diagonal-Matrix mit Diagonaleinträgen 3.
- Das Produkt zwischen Matrizen und/oder Vektoren ist mit einem schlichten Punkt “.” gekennzeichnet.

¹bei Fragen: anegrett@physik.uni-hamburg.de