Elementary decomposition of the Grover diffusion operator and applications

Sebastian Miles* and Elliot Bryce†

(Dated: January 23, 2020)

Elementary gate decompositions are studied in order to find a decomposition of Grover's diffusion operator, in the following denoted by $\mathcal{D}$. A method is described with which this decomposition can be generalized to arbitrary numbers of qubits, a number which is in the following text denoted by n. The decomposition was implemented into `QuTip` to test the results of [1] and [2] in particular the properties of the diffusion operator for arbitrary probability distributions of the input. The predictions of [1] and [2] could be recovered for both uniform and non uniform distributions.

I. INTRODUCTION

In order to make sophisticated quantum algorithms accessible for current day quantum computing hardware one needs to break done the, in general, complex functions implemented in quantum algorithms into smaller bits such that today's technology can realize them probably. By today's standard this means that we need to find decompositions into single or at most 3-qubit gates to realize the algorithm. In this report we want to investigate the decomposition of Grover's diffusion operator defined by

$$\mathcal{D}_{i,j} = \begin{cases} 2/N - 1 & i = j \\ 2/N & i \neq j \end{cases}$$

with $N = 2^n$, realizing the mathematical operation of an inversion of a general probability distribution around its mean.

This operator $\mathcal{D}$ can be used very generally in its purpose therefore it is not surprising that if in Grover's algorithm the oracle step is omitted the results can still be of use in a more general algorithm, such as a subroutine of a larger process. This behaviour has been investigated for over 20 years, leaving us with a theoretically good understanding of the operators properties. However, this behaviour has, to the authors best knowledge, not yet been observed either in a simulation or real machine. The simulation aspect will be the subject of this paper.

II. DECOMPOSITION

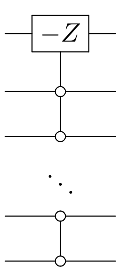
Following Grover's original construction [3] the diffusion operator can be decomposed into operations $\mathcal{D} = \mathcal{W}\mathcal{S}\mathcal{W}$ such that $\mathcal{W} = \mathcal{H}^{\otimes n}$ and $\mathcal{S}$ can be written as

$$\mathcal{S}_{i,j} = \begin{cases} 1 & i = j = 1 \\ -1 & i = j \geq 1 \\ 0 & i \neq j \end{cases}$$

The $\mathcal{W}$ operator already has a form that is implementable so its consideration will be omitted. The interesting operator in this decomposition is the $\mathcal{S}$ operator that adds a phase of π to only the $|0\rangle^{\otimes n}$ state and leaves all others invariant. The first choice to make this problem more accessible is to use the freedom of a global phase that one has to pull out a phase of π. This inverts the signs of the matrix elements of the operator leaving only the mentioned $|0\rangle^{\otimes n}$ state to be phase inverted. This choice is useful as the problem was now reduced to solving the controlled inversion only on one of the possibly many qubits. The operator could now be realized by a Pauli-Z gate on the first qubit that is controlled by all other qubits being in zero as depicted in the circuit diagram below.

It has been shown by Barenco et al. [4] that the set containing only CNOT and single qubit rotation gates is universal in a sense that every unitary can be realized with arbitrary precision while only utilizing these gates. By realizing that a given circuit can be decomposed using its associated truth table, meaning which gates should be controlled by a set of others at a given step, the decomposition of the $\mathcal{S}$ operator can be realized in this set. To do that, consider the n-Bit decomposition equation in [4] that describes the truth table decomposition of a given gate. This decomposition makes use of fractions and inverses of fractions of the gate that is to be implemented. To make use of this equation one needs a diagonalisation and associated basis transformation because the gate fractions can then be easily realised.

Using the simplifications applied above, the task is to find an elementary gate decomposition of



Where

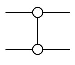
$$\boxed{-Z} = \begin{pmatrix} e^{i\pi} & 0 \\ 0 & e^{i2\pi} \end{pmatrix} \qquad (1)$$

Recalling the operator for two qubits we can directly

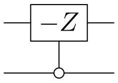
[*] s.miles@student.tudelft.nl

[†] e.j.f.bryce@student.tudelft.nl

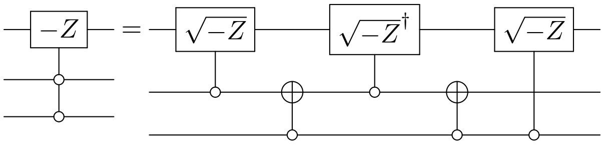
apply this notion



$$= \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \qquad (2)$$



Clearly, the equality on the left holds as well when one considers the gates actions on the individual states. Using Barenco et al.s‘ insights, this can be generalized to more qubit variants. The three qubit decomposition is for example



Where

$$\boxed{\sqrt{-Z}} = \begin{pmatrix} e^{i\pi/2} & 0 \\ 0 & 1 \end{pmatrix} \qquad (3)$$

To proof the claim one can look at the individual gate matrices from which the complete gate is composed. Doing this will proof the claim riguorously. A more intuitive proof can be derived by considering how the low controls are used in the decomposition. The gate is only applied when $q_1 \wedge q_2$ are zero, but not when $q_1 \vee q_2$ is zero thereby effectively realizing the desired identity $q_1 \wedge q_2$ up to a power of the controlled gate.

Bigger system sizes can be tackled either by using the above relation iteratively where the two controls can be any pair of subdivisions of a larger system, or by utilizing the same rational of the explanation and computing a Gray code decomposition following

$$\begin{aligned} &\sum_{k_1} q_{k_1} - \sum_{k_1 < k_2} (q_{k_1} \oplus q_{k_2}) + \sum_{k_1 < k_2 < k_3} (q_{k_1} \oplus q_{k_2} \oplus q_{k_3}) \\ &\quad - ... + (-1)^{m-1}(q_1 \oplus ... \oplus q_n) \\ &\quad = 2^{m-1} \times (q_1 \wedge q_2 \wedge ... \wedge q_n) \end{aligned} \qquad (4)$$

from [4]

III. GENERALIZED GROVER ALGORITHM

Since the initial introduction to quantum oracle search algorithms due to Grover [3] some effort was put into exploring generalizations and optimal points of effectiveness of this technique. This paper explores the most general algorithm suggested by Biham et al. [1] testing their and associated claims on the generalized Grover algorithm [2]. The goal was to rediscover their given results regarding maximum probability of success and evolution of the averages.

The generalization that we consider allows predictability of the averages and variances of marked and unmarked states that follow an arbitrary amplitude distribution. Say the system has N states of which r states are marked and $N - r$ states are unmarked. Then, for $r \in [1, N/2]$ the opitmal measurement times are

$$T = \frac{(j + 1/2)\pi - \arctan\left[\frac{k\bar{(0)}}{l\bar{(0)}}\sqrt{r/(N-r)}\right]}{\arccos\left[1 - 2r/N\right]} \qquad (5)$$

for $j \in \mathbb{N}_0$, $k\bar{(0)}$ being the average of the marked input states, as well as $l\bar{(0)}$ the average of the unmarked input states [1]. Also, the maximum success probability is predictable as

$$P_{max} = 1 - (N - r)\sigma_l^2 \qquad (6)$$

with σ_l the variance of the unmarked states.

These two results can be tested using simulation packages such as `QuTip` or `ProjectQ`. However, a major problem regarding efficiency of elementary gate decompositions is the number of gates needed to achieve the desired gate. For the set consisting of single qubits and CNOTs it can be proven (see for example [5] and [6]) that the optimal decomposition of an arbitrary gate is $\mathcal{O}(4^n)$. This exponential growth leads to very high computational demands when one wants to simulate the decomposition using tools such as `QuTip` or `ProjectQ`. To cope with the performance demand but still have at least a minimal statistic we chose to restrict our circuits on $N \leq 5$.

IV. IMPLEMENTATION

A. Code Structure

To verify the accuracy of the decomposition, a script was developed using `QuTip` to simulate the general Grover search algorithm with an arbitrary input distribution. The properties of this algorithm were analysed by testing against Eq. (5) and Eq. (6). The $N = 5$ decomposition (see App. A) was implemented into `QuTip` as the sequence of the individual gates resulting in matrix associated with the operator. This matrix could then be used for both the diffusion operator and the oracle. To provide freedom in which states to mark, we made the decomposition initially with controls to be active when receiving a ’1’ input. Then the diffusion operator was found by applying an $X^{\otimes 5}$ gate (for 5 qubit register) so that the controls were all now active for a low input. The Oracle could be found similarly, with a U gate where U is formed of some tensor product combination of X, I gates on each qubit. The configuration depended on what the requested solutions to mark where. This is demonstrated in Fig. 1.

B. Optimal Number of Steps

The average amplitude for the marked states after each iteration was plotted against the iteration number to pro-

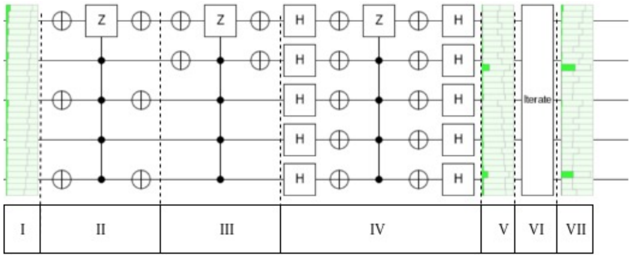


FIG. 1. The geometry of the simulated circuit. (I) Here the input distribution is arbitrary. (II) This is the first part of the oracle, where the first of the two marked solutions is 'marked'. The surrounding X gates configuration set this marked solution as $|01010\rangle$. (III) Here marked solution is $|00011\rangle$. (IV) Diffusion operator including the 5 qubit Hadamard transform surrounding the conditional phase. (V:VII) marked solution increase in probability after a second iteration of the step from II:IV.

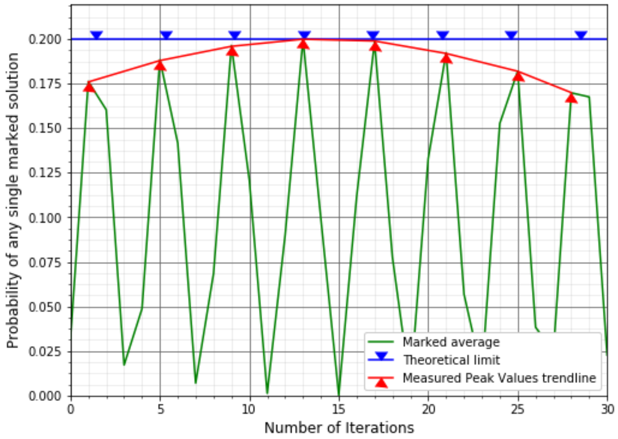


FIG. 2. Comparison of measured iterations of local peak probability against where theoretically the peak should be at the theoretical limit. As the difference between measured T and theoretical T decreases, the probability increases. Here we can see the red trend-line is asymptotic with the theoretical value which is a good indication that that out simulation behaves according to (5) despite being limited by integer iterations. The initial distribution used to generate this plot was uniform.

duce Fig. 2. Due to our limitation to a 5 qubit system the optimal measurement times predicted by Eq. (5) could not be reached as the predicted values are not integer values. A possible solution by interpolation of the points neighbouring the theorised peak is also of limited use due to the small sample size. The method used was to extrapolate the envelope frequency of the measured peak values which arises from the iteration term, $j = 0, 1, 2..$, in (5).

$$T = \frac{j\pi}{\arccos\left[1 - 2r/N\right]} \tag{7}$$

An alternative way of looking at this approach is, by plotting many peaks, the chances one of those theoretical values will be close to an integer. This envelope can be seen in Fig. 2.

The same technique can also be applied for multiple marked states. However, the probability shows, as to be expected, to be lower. Combining the probability of all marked solutions will achieve a max probability equivalent to one as for a uniformly distributed input.

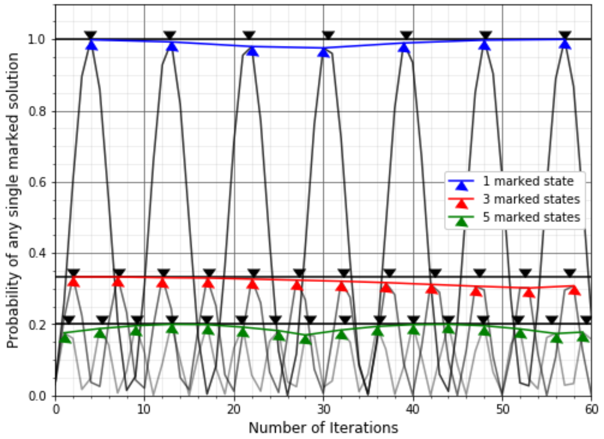


FIG. 3. Effect on the probability oscillations of a single marked solution compared for different numbers of total marked solutions. It is important to note that here, $\sigma_l^2 = 0$, which will become apparent in Fig. 4. With that it can clearly be seen that for 5 solutions with zero variance, all simultaneously have a peak probability of $P = 0.2$. So the combined probabilities are $0.2 * 5 = 1.00$.

C. Variance on theoretical limit

To verify the general case for (6), the system requires some variance so that $\sigma_l^2 \neq 0$. To do this randomness was introduced into the distribution order the increase the variance of the distribution. The effect of introducing this randomness is seen in Fig. 4. Significantly the theoretical limit falls as the variance increases. As with all previous figures, the theoretical limit was calculated before starting the iteration and it only requires the initial amplitude distribution. Although randomness is introduced we can see that the simulation recovers the theoretical prediction up to a deviation due to integer iterations. This is a significant result as it shows that, indeed, the behaviour of Grover's generalized algorithm is predictable and produces forseeable results. Furthermore, Fig 4 shows that also the optimal measurement times, that have been predicted by Eq. (5), can be confirmed by the simulation. Although the optimal measurement times have not necessarily been hit while doing integer iterations, still the probability to recover the marked states significantly increased.

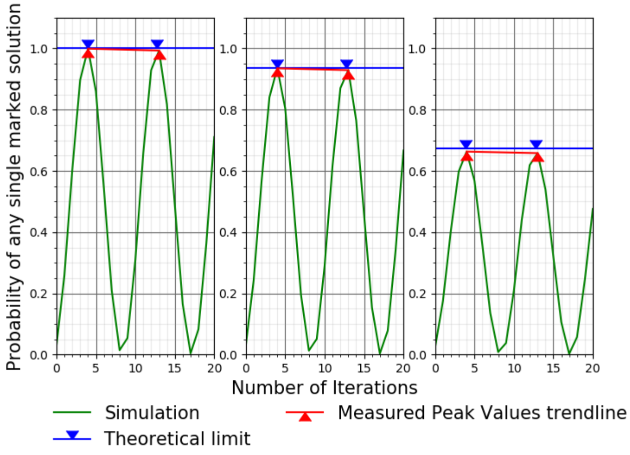


FIG. 4. (Left) $var = 0$, (Centre) $var = 0.2$ and (Right) $var = 0.4$. As variance increase there is a significant drop in the theoretical limit, which is also found in the simulation results calculation. var was a term used in the script that build up the arbitrary non uniform input. (See appendix B

V. DISCUSSION

Figure 2 demonstrated that the envelope function is useful in finding the optimal values but for a direct measurement we can see how close simulation gets in Fig. 5. At the very least it can be said that all measured peaks are within a 0.5 distance from the accompanying theory value and so we are limited by resolution rather than any systematic error.

Figure 3 demonstrated the effect of number of solutions on the probability oscillations. Aside from the previously mentioned, there is another notable feature seen. The envelope function oscillates faster as the number of solutions increase. This behaviour was to be expected following Eq. (5). Taking r to be larger leads to a decrease of $\Delta T = T_{n+1} - T_n$ and a therefore higher oscillation frequency.

For Fig. 4 an error was encountered for the variance measurements, and the calculated values where all tiny in comparison to the spread of the individual marked state oscillations. When attempting to add error bars, they were too small to be clearly visible. Adding to this the asymtopic relationship seen would occasionally fail.and the measured averages would be significantly above or below the theoretical limit.

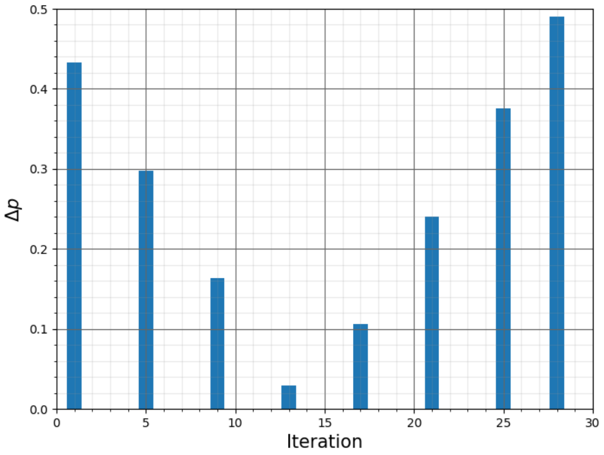


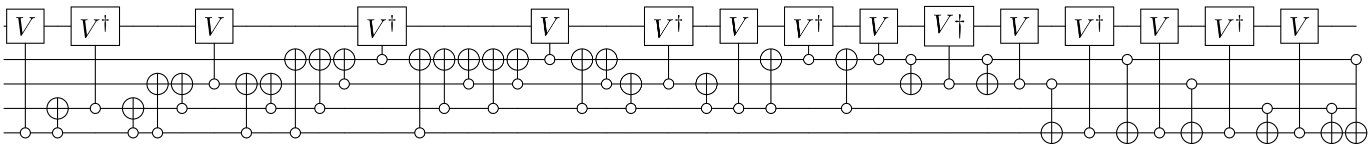
FIG. 5. Relating to Fig. 2 bar heights show the difference in iteration between simulation and theory, significantly all are below a difference of 0.5.

VI. CONCLUSION AND OUTLOOK

We have shown that the central results presented in [2] and [1] do indeed hold and that the introduction of variance into the input distribution still leads to controllable and predictable behaviour. The shown decomposition can in the near future also be of use in experimental setups with limited sets of gates in order to benchmark the hardware while still having a forseeable behaviour. Future research into this topic could attempt to implement the demonstrated algorithm on experimental hardware in order to validate the results in a real setup. The effect of noise on the algorithm's behaviour can then be analyzed. Furthermore, attempts can be made to generalize the given strategy of the decomposition in an explicit algorithm if this is not yet existent.

Appendix A: Circuit diagram of the implemented gate

To make the rationale of the decomposition clear and to precisely show what was implemented into QuTip does the following circuit diagramm show the full decomposition of the gate.



Where

$$V = (-Z)^{1/8} = e^{i\pi} \begin{pmatrix} e^{i2\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{pmatrix} \tag{A1}$$

This decomposition that is still uncompiled is the basis

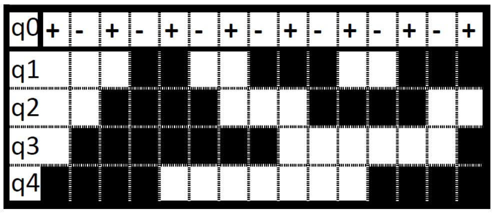


FIG. 6. Depiction of the Gray code that is associated with the decomposition. The black squares refer to quantum wires that are successively applied and unapplied on one of the wires to control the gate in q_0. The top row refers to the usage of V on a plus and $V^\dagger$ on a minus.

for our implementation. It realizes the Gray code decomposition that is shown in Fig. 6.

Appendix B: Function to create the non-uniform input distribution.

Randomness is built from uniform.random command. The individual 2 level qubits are first created and then each multiplied by their random matrix:

$$\begin{pmatrix} rand & 0 \\ 1-rand & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} rand \\ 1-rand \end{pmatrix} \tag{B1}$$

The randomised qubits are all combined with the tensor product. *rand* is found using the command `np.random.uniform(0.5-var, 0.5+var, 5)` where *var* is given as an input.

[1] Eli Biham Markus Grassl Daniel A. Lidar David Biron, Ofer Biham. Generalized grover search algorithm for arbitrary initial amplitude distribution. *Physical Review A*, 60(2742), 1998.

[2] Peter Høyer Alain Trapp Michel Boyer, Gilles Brassard. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4-5), 1999.

[3] Lov K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, 79(325), 1997.

[4] Richard Cleve David P. DiVincenzo Norman Margolus Peter Shor Tycho Sleator John A. Smolin Harald Weinfurter Adriano Barenco, Charles H. Bennett. Elementary gates for quantum computation. *Physical Review A*, 52(3457), 1995.

[5] Martti M. Salomaa Juha J. Vartiainen, Mikko Möttönen. Efficient decomposition of quantum gates. *Physical Review Letters*, 92(177902), 2004.

[6] Juha J. Vartiainen Mikko Möttönen. Decompositions of general quantum gates.