KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Computer Engineering

First cycle, 15 credits

Impact of noise in amplitude amplification on IBM Q

EDVARD ALL
DILVAN SABIR

Stockholm, Sweden 2022

Impact of noise in amplitude amplification on IBM Q

EDVARD ALL

DILVAN SABIR

Bachelor in Computer Science
Date: June 8, 2022
Supervisor: Stefano Markidis
Examiner: Pawel Herman
School of Electrical Engineering and Computer Science
Swedish title: Påverkan av störningar för amplitudamplifikation på IBM Q

Abstract

Quantum computers promise immense speed up compared to even the most advanced super-computers of today, but environmental noise introduces errors in the former, constituting a significant hurdle for realising these promises. Quantum error-correcting codes have been suggested as a strategy to alleviate this hurdle, including codes optimised towards computation under specific models of noise. Investigating the impact different types of quantum error have on the results of quantum computation could therefore be instructive for the construction of these optimised quantum error-correcting codes.

We simulate diffusion, a central operation in an important quantum algorithm, Grover's algorithm, concerning different noise conditions. Our results are inconclusive with respect to discerning the difference between different types of noise and their effect on the output of this operation, but are in line with previous research in that errors in even five qubit circuits degenerate the results to the point where they are by and large unusable.

Sammanfattning

Svenska abstraktet ska in här.

Contents

Chapter 1

Introduction

Quantum computing entails harnessing various properties of quantum mechanics, such as superposition, interference, and quantum entanglement, to perform calculations in a parallel manner not possible on classical systems. Assuming a closed system, quantum technology entails great promises regarding what problems may be deemed feasibly computable. [1]

Unfortunately, the assumption of a closed system is not realistic – real systems are, without exception, part of an environment and consequently subject to noise. This noise remains a significant hurdle in the road toward building large-scale quantum computing devices – solving problems of non-trivial size – as the impact of noise (and therefore number and gravity of errors) scales with the size of the quantum computing device [2, p. 9].

Noise in quantum computation stems from e.g. low fidelity of the quantum hardware and environmental noise such as thermal and gravitational perturbation [3, p. 354], [4]. Although this noise may be reduced, it is reasonable to expect that the environmental noise introduced by control operations and output measurement will not be resolved in the foreseeable future. Various strategies for coping with this noise have therefore been suggested, a prominent example of such a strategy being the use of quantum error-correcting codes (QEC) (see [5, 6, 7, 8]).

QECs, like their classical counterpart, introduces overhead in the form of an additional amount of required physical qubits. To alleviate this fact, QECs tailored for specific models of noise have been studied and found to outperform generic ditto with regards to efficiency [9, 10, 11]. Thus, understanding the nature of noise models, how they affect different quantum circuits, and how accurately they map to real, open quantum systems is useful for the proper application of such QECs.

1.1 Problem definition

Efficient fault tolerance for quantum computing devices is instrumental if actual quantum speedup is to be attained. To efficiently make use of quantum error-correcting codes, it is necessary to understand how models of noise map to errors in quantum computation. This project aims to investigate the behaviour of a prominent quantum algorithm, Grover's search algorithm, as subject to various models of noise. To this aim we investigate the following question:

- Do the results of Grover's algorithm significantly vary under different models of noise?
- Under what conditions are results no longer usable (in the sense that the correct solution may be discerned)?

1.2 Scope and constraints

We aim to achieve this through the use of quantum machines for our experiments, made available by the IBM quantum cloudware platform [12]. However, given the limited availability of such machines, we rely on simulating quantum computation (and mentioned noise models). Specifically we will use up to five qubits, simulating the IBMQ-lima (falcon r4t).

Furthermore, because noise and error may manifest in numerous different ways, we investigate results under different simulated conditions (with respect to noise). These conditions relate to number of qubits in the circuit, as well as the channels of noise affecting the qubits. Because we simulated circuits of different numbers of qubits, and as oracle functions may be constructed in a number of different ways, we chose to omit the oracle function (i.e. phase shift operation of Grover's) as it could obfuscate error analysis.

1.3 Thesis overview

1.4 Acknowledgments

We acknowledge the use of the IBM Q environment for this work. The views expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Q team.

Chapter 2

Background

2.1 Quantum bits

A quantum bit or *qubit* is a unit of information that describes a two-dimensional (or two-state) quantum mechanical system and constitutes the most basic unit of quantum information [13, p. 139]. The qubit is somewhat analogous to the classical bit. However, unlike the classical bit, which can only be in one of two discrete states (represented as 0 or 1) at any given point, quantum mechanics allow for the qubit to be in a *superposition* of both states simultaneously. This implies that qubits don't have a discrete value but rather a probability to collapse to a specific state upon measurement.

A qubit may then be represented as a complex 2-by-1 matrix $\begin{bmatrix} c_0 & c_1 \end{bmatrix}^T$, where $|c_0|^2$, $|c_1|^2$ denotes the probability of the qubit collapsing to state $|0\rangle$, $|1\rangle$ respectively, upon measurement [13, p. 117]. In other words, the state of a qubit can be described by the linear combination $|\psi\rangle = c_0 |0\rangle + c_1 |1\rangle$.

By representing these probabilities as complex numbers we are able to model the notion of quantum interference, whereby a quantum particle exhibits the wave-like property of two probabilities cancelling each other out (much like two waves colliding may cancel eachother).

We may also represent the state of the qubit geometrically, as a unit vector in $\mathbb{R}^3$, outlining what is known as the Bloch sphere (see figure 2.1). When we measure the qubit and it collapses to state $|0\rangle$ or $|1\rangle$, we may view it as the qubit vector pointing to the north and south pole respectively. In this sense the probability of the qubit collapsing to either state is determined by the latitude (its angle θ) of its unit vector in the Bloch sphere.

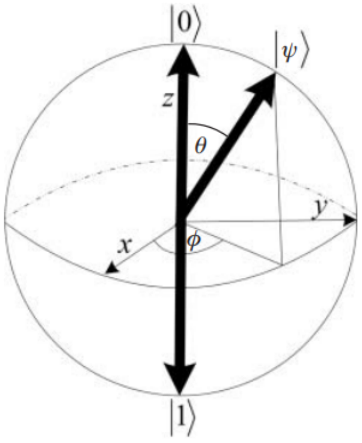

Figure 2.1: A geometric representation of a qubit, the Bloch sphere [13, p. 161]
.

2.2 Quantum gates

Like the classical gate operates on a classical bit, so do quantum gates operate on qubits. However, quantum gates differ from their classical counterpart in being reversible, a constraint that follows from the unitarity principle in quantum mechanics. The only exception to this reversibility is measurement.

A quantum gate may then be represented by some unitarian 2^n-by-2^n matrix, operating on n qubits. Take the set of Pauli-gates for example (where the X-gate corresponds directly with the classical NOT-gate):

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Notably, we may view these operations on a qubit as rotations of its Bloch sphere, where X, Y, Z flip the Bloch sphere half-way round the corresponding axis.

2.3 Noise in quantum computing

Real world quantum devices are subject to environmental noise, and so do not conform to the closed model outlined thus far. This environmental noise affects and distorts the quantum information within the system – the quantum state degenerates from a pure to a mixed state – potentially causing errors in computation. This is what is referred to as *decoherence*.

2.3.1 Decoherence

Decoherence is the coupling of a quantum system to its external environment, leading to an increase in entropy within the system. This loss of information interferes with our assumptions of how the system would behave were it not for this decoherence, and so strategies to alleviate this problem have been suggested. Two such strategies are:

Fast gates execution: if gates in the circuit execute sufficiently fast, then decoherence is not a problem. This strategy might not be sufficient however, as it may well be the case that gates cannot satisfy this criteria.

Fault-tolerance may be achieved by employing quantum error-correcting codes [13]. Just as classical computing may be rendered fault-tolerant by way of error-correcting codes, quantum error-correcting codes have been developed and in turn given grounds for a theory of fault-tolerant quantum computing [2, p. 10]. Although promising, the practical application of this theory is constrained by the fact that such error codes introduce significant overhead, requiring additional qubits [1].

2.4 Grover's Algorithm

Grover's algorithm is a quantum algorithm that, given some function $f : \{0,1\}^n \to \{0,1\}$ such that $f(x_0) = 1$ for some x_0, and 1 otherwise, finds such an x_0 [14]. Because we are inverting the function in this manner, the algorithm may be used to find some desired element in an unordered set, if we are able to construct an f such that f produces some value for our desired element x_0 and another value for all other elements. As such, it is a prominent algorithm potentially useful for a number of different related problems, such as boolean satisfiability problems,

Grover's algorithm makes use of the general quantum method of amplitude amplification to boost the probability of measuring a desired state, and can be divided into four stages (where stage 3 constitutes this amplitude amplification):

1. Start with some state $|0\rangle^{\otimes n}$, n being the number of qubits.

2. Apply $H^{\otimes n}$, thereby putting the system in a superposition.

3. Repeat $\sqrt{2^n}$ times:

   (a) Using the oracle function f, flip the phase of the desired state(s).
   (b) Invert phase of the state about the mean (commonly referred to as the diffusion operation D).

4. Measure the system.

Note that n qubits represent some $2^n = N$ states. The algorithm is in $\mathcal{O}(\sqrt{N})$ time complexity, where N is the number of elements in the set, which is also the lower bound of quantum search of an unordered set [15]. This analysis assumes the oracle function is given to us.

2.4.1 Constructing the diffuser

[In this section we explain the diffusion operation and how its circuit may be constructed.]

2.5 Previous work

Mandviwalla, Ohshiro, and Ji [16] explore the feasibility of Grover's algorithm by way of IBM Q, concluding that at the time of the study's publication, quantum computers could only accurately and reliably solve simple problems with small amounts of data. Additionally, their experimental results suggest that quantum computers are slower than expected.

Wang and Krstic [17] explore error thresholds for Grover's implementations of three different characteristics: (1) a standard implementation, (2) an implementation of reduced circuit depth, and (3) the former with and without ancilla bits on the multi-control Toffolis. They find that reduced circuit depth does indeed lead to higher error thresholds (i.e. better robustness to noise), which is further improved by the use of an ancilla qubit.

Chapter 3

Method

3.1 Qiskit

Qiskit is an end-to-end open-source framework for quantum computing developed by IBM, encompassing: low-level hardware interaction, simulation, emulation, all the way to the higher-level abstraction of applied algorithms [18]. Qiskit thereby allows cloud access to IBM quantum hardware [19].

In the case of simulation, the Qiskit Aer API [20] furthermore provides noise models, useful for studying the behaviour of algorithms in the inevitable presence of quantum circuit error. Qiskit documentation is available online at `https://qiskit.org/documentation/`.

3.2 Simulation

We constructed diffusers of size 2, 3, and 5 qubits (see fig 3.1), and ran simulations of these under several noise profiles 100'000 times.

3.2.1 Nosie profiles

The following noise profiles were introduced to the simulated systems:

- No noise (Baseline)
- Realistic noise (Real)
- Depolarizing errors only (Depolarizing)
- Amplification damping errors (Amplification)
- Phase Damping errors (Phase)

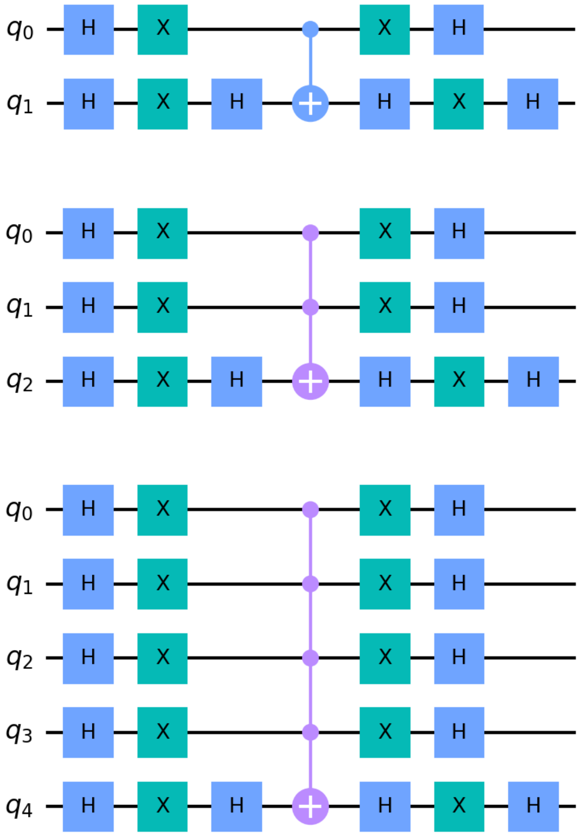

Figure 3.1: Circuits for diffuser of size two, three, and five qubits respectively.

The no-noise variant was our established baseline profile, which we expect to be a uniform distribution as the diffusion is a reflection about the mean, and the reflection about a uniform superposition $|s\rangle$ is just $|s\rangle$ itself.

The realistic noise model was gathered from the IBMQ-lima backend. The parameters for the custom noise profiles were based on parameters from the IBMQ-lima backend (see Appendix B), to ensure that simulated conditions were reasonably realistic: a rough mean of the qubit error was appplied to all qubits. No readout error was applied, however.

3.3 Evaluation

To evaluate the resulting distributions we calculated the mean squared error (MSE) of each respective noise profile with regards to the baseline.

Chapter 4

Results

[Because we realised we had to change our distribution comparison too late, we did have time to write up these results properly]

As can quickly be gathered from observing heatmap figures in 4.1, errors vastly increase under the realistic noise model. From the values reported in table 4.1, we calculated an MSE of [NOT YET CALCULATED], between this real noisy distribution and the clean baseline on the two and three qubit circuits respectively.

As for the distributions of the custom noise profiles, no significant discrepancy may be discerned.

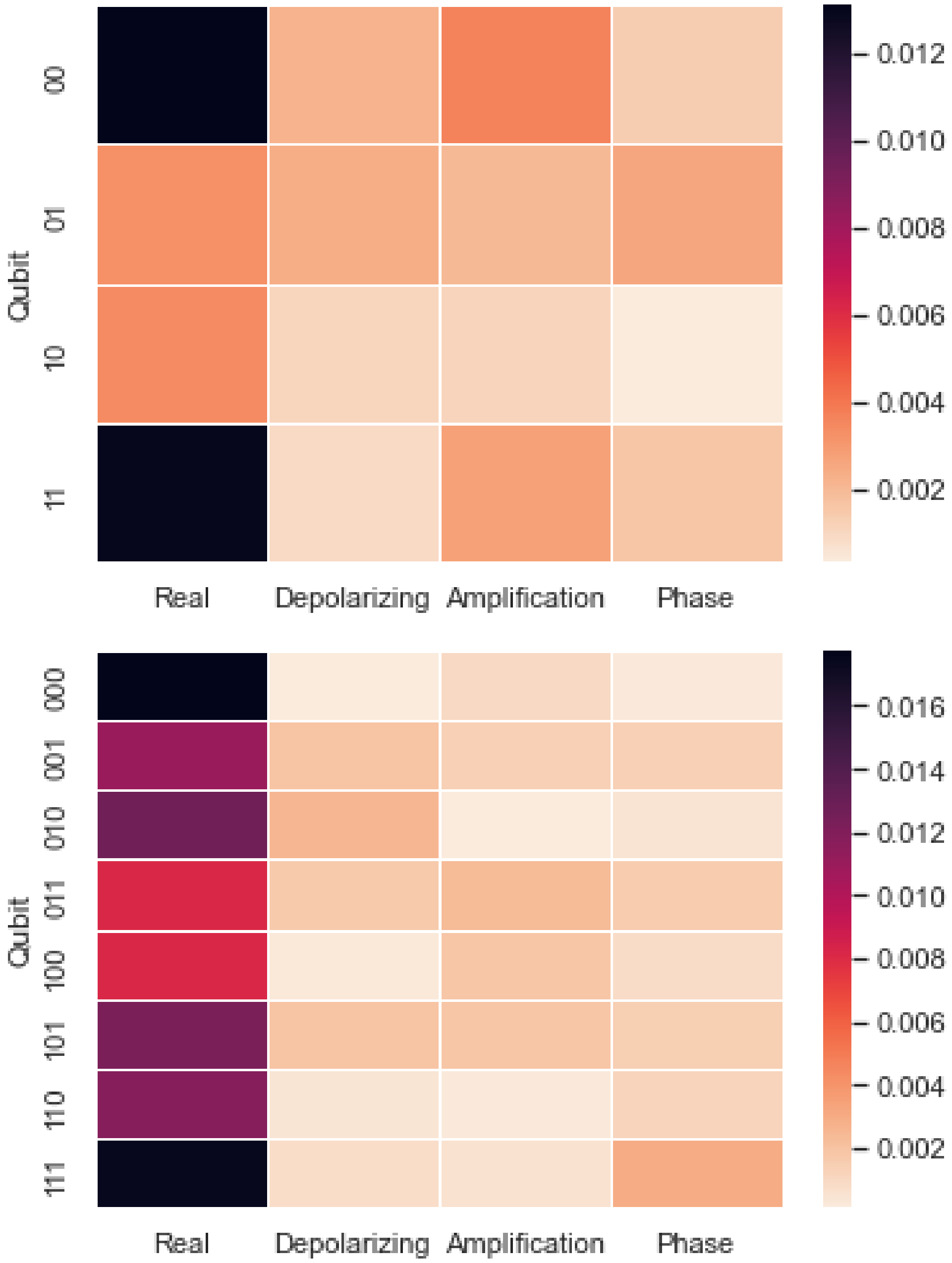

Figure 4.1: Heatmap over the proabability map over the states of 2 and 3 qubits after 100 000 runs as compared to baseline.

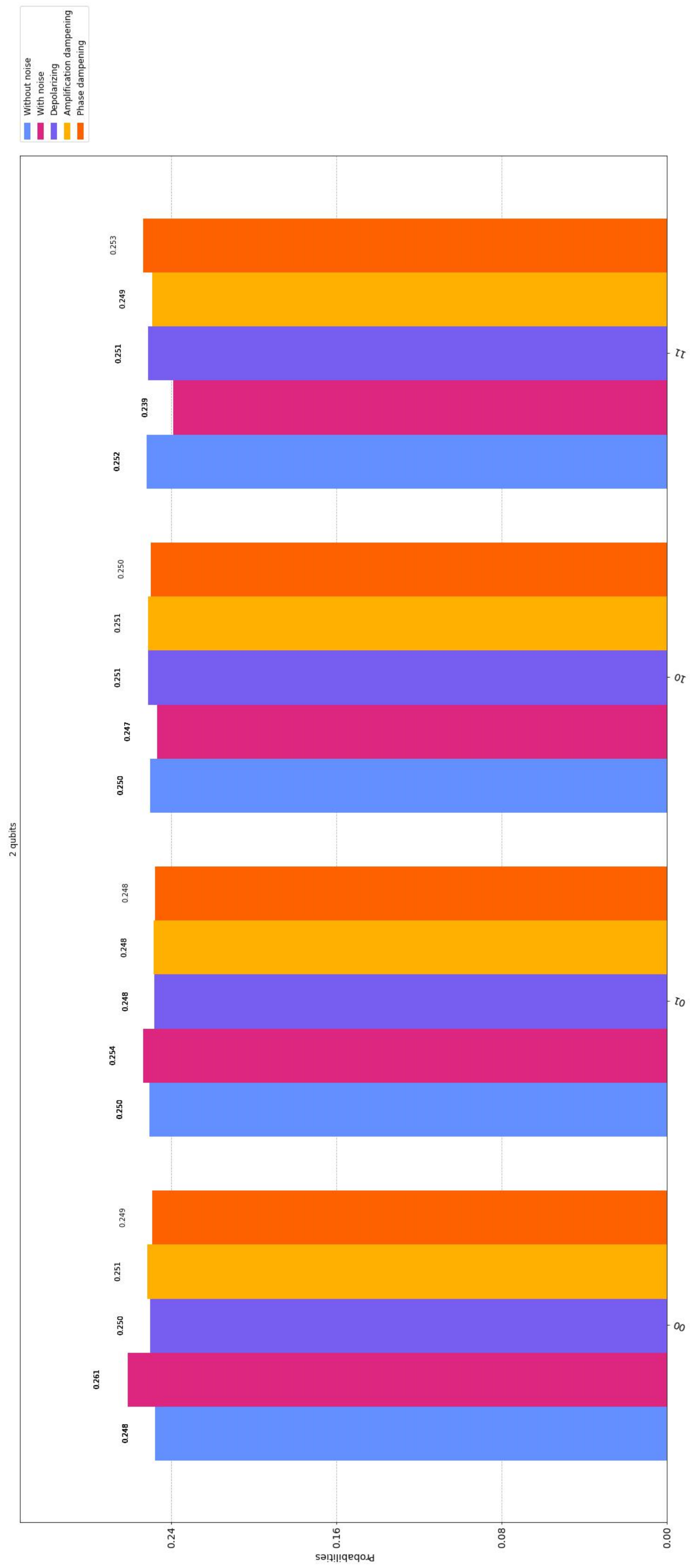

Figure 4.2: Histogram over the states of 2 qubits after 100 000 runs.

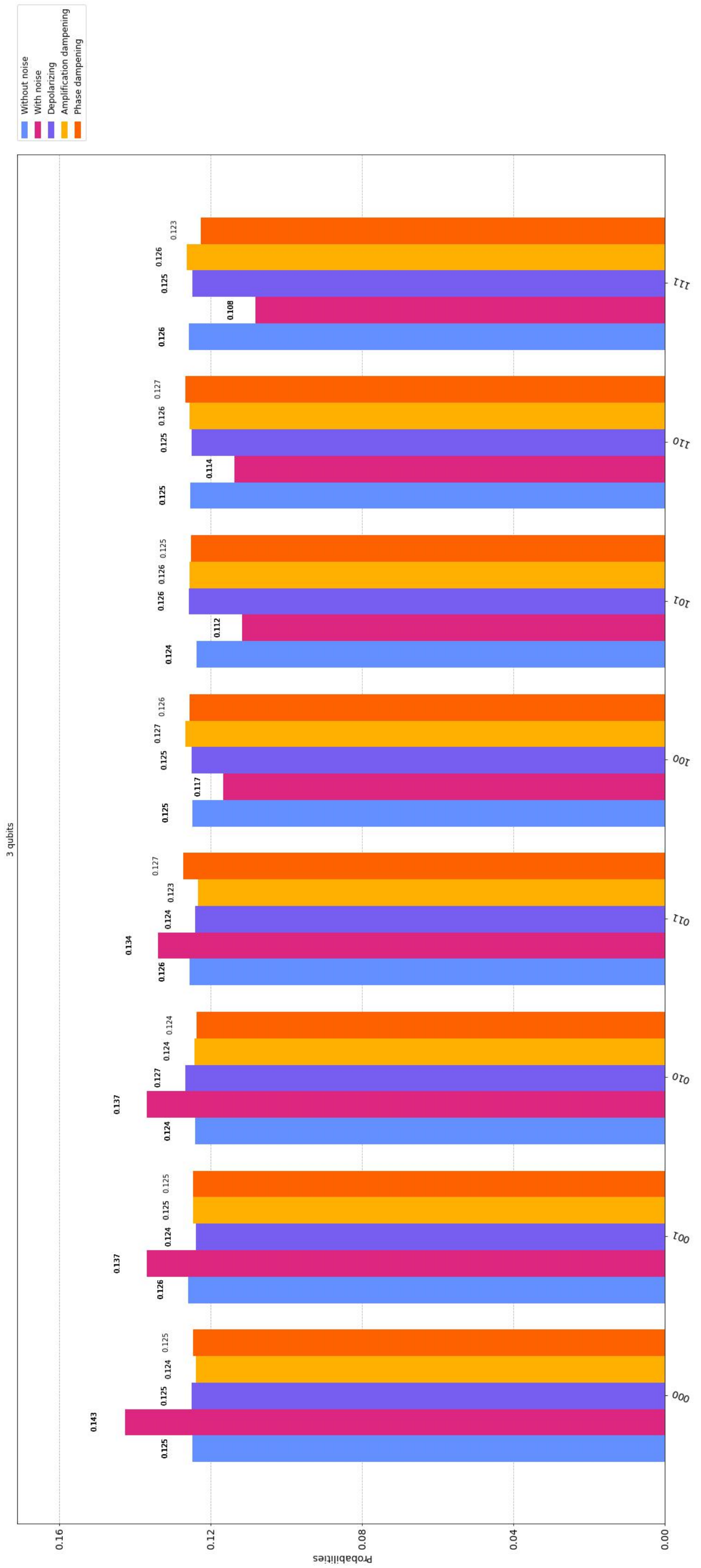

Figure 4.3: Histogram over the states of 3 qubits after 100 000 runs.

Qubit	Baseline	Real	Depolarizing	Amplification	Phase
00	0.24780	0.26090	0.24999	0.25144	0.24913
01	0.25030	0.25353	0.24793	0.24831	0.24771
10	0.25013	0.24672	0.25119	0.25121	0.24980
11	0.25177	0.23885	0.25089	0.24904	0.25336

Qubit	Baseline	Real	Depolarizing	Amplification	Phase
000	0.12482	0.14258	0.12493	0.12388	0.12461
001	0.12591	0.13686	0.12394	0.12453	0.12453
010	0.12412	0.13679	0.12665	0.12428	0.12363
011	0.12560	0.13377	0.12396	0.12328	0.12721
100	0.12474	0.11660	0.12502	0.12659	0.12557
101	0.12376	0.11157	0.12567	0.12561	0.12521
110	0.12542	0.11363	0.12499	0.12562	0.12661
111	0.12563	0.10820	0.12484	0.12621	0.12263

Table 4.1: Probability distribution of different qubit states in the different profiles for two and three qubits.

Qubit	Baseline	Real	Depolarizing	Amplification	Phase
00000	0.3175	0.4083	0.3177	0.3182	0.3115
00001	0.3059	0.3584	0.3121	0.3133	0.3040
00010	0.3156	0.3359	0.3157	0.3249	0.3087
00011	0.3145	0.2930	0.3035	0.3230	0.3074
00100	0.3119	0.3761	0.3170	0.3131	0.3146
00101	0.3185	0.3390	0.3099	0.3169	0.3169
00110	0.3123	0.3285	0.3095	0.3090	0.3143
00111	0.3116	0.2790	0.3125	0.3005	0.3062
01000	0.3179	0.3548	0.3088	0.2983	0.3110
01001	0.3171	0.3152	0.3049	0.3152	0.3091
01010	0.3068	0.2936	0.2972	0.3177	0.3148
01011	0.3104	0.2485	0.3156	0.3136	0.3041
01100	0.3037	0.3461	0.3051	0.3141	0.3097
01101	0.3133	0.3096	0.3109	0.2969	0.3226
01110	0.3181	0.2895	0.3167	0.3164	0.3122
01111	0.3158	0.2461	0.3087	0.3090	0.3159
10000	0.3178	0.4049	0.3055	0.3079	0.3137
10001	0.3125	0.3385	0.3119	0.3174	0.3252
10010	0.3153	0.3123	0.3239	0.3165	0.3119
10011	0.3127	0.2738	0.3151	0.3075	0.3154
10100	0.3171	0.3749	0.3150	0.3142	0.3132
10101	0.3046	0.3132	0.3134	0.3164	0.3170
10110	0.3094	0.3021	0.3160	0.3231	0.3068
10111	0.3066	0.2471	0.3173	0.3091	0.3112
11000	0.3121	0.3489	0.3137	0.3150	0.3088
11001	0.3100	0.3048	0.3163	0.3080	0.3107
11010	0.3055	0.2850	0.3206	0.3056	0.3074
11011	0.3098	0.2428	0.3211	0.3078	0.3086
11100	0.3081	0.3321	0.3046	0.3093	0.3212
11101	0.3234	0.2842	0.3053	0.3224	0.3141
11110	0.3112	0.2778	0.3197	0.3100	0.3150
11111	0.3130	0.2360	0.3148	0.3097	0.3168

Table 4.2: Probability distribution of different qubit states in the different profiles for 5 qubits.

Chapter 5

Discussion

As can be plainly observed already in comparing figure 4.2 and figure 4.3, variance of the resulting distributions increases dramatically as the number of qubits in the circuits increase. At five qubits, the distribution degenerates from the desired uniformity as to make discerning the desired state impossible.

Comparing the distributions of the different noise models suggests that the type of error matters little with regards to the output distribution. There are however a number of limitations to our method that makes it difficult to draw clear conclusions from the results. Our custom noise profiles were simplified for reasons of transparency and easy of implementation, which might have rendered them overly optimistic. In line with this decision, readout error was not included, under the assumption that this error could just as well be tacked on to the results ad-hoc. Furthermore, we constrained simulation to the diffusion operation of Grover's algorithm, thus simplifying the circuit and lowering the accumulated number of errors of the output state.

Therefore, further studies could improve on these noise models, as to ascertain conclusive results regarding the effect various noise has. Additionally, oracles could be constructed and introduced to the circuit, whereby the errors of this circuit could be compared to the oracle-less one.

5.1 Conclusions

Our results suggest that the type of quantum error matters little with regards to the uniformity of the resulting distribution, although these results should not be considered conclusive. On a five-qubits circuit, the diffusion operation is already so unstable as to make execution of Grover's algorithm in its entirety fruitless.

Bibliography

[1] John Preskill. “Quantum computing in the NISQ era and beyond”. In: *Quantum* 2 (2018), p. 79.

[2] Dave Morris Bacon. *Decoherence, control, and symmetry in quantum computers*. University of California, Berkeley, 2001.

[3] Michael A Nielsen and Isaac Chuang. *Quantum computation and quantum information*. 2002.

[4] Angelo Bassi, André Großardt, and Hendrik Ulbricht. “Gravitational decoherence”. In: *Classical and Quantum Gravity* 34.19 (2017), p. 193002.

[5] Isaac L. Chuang and Yoshihisa Yamamoto. “Quantum Bit Regeneration”. In: *Phys. Rev. Lett.* 76 (22 May 1996), pp. 4281–4284. DOI: 10.1103/PhysRevLett.76.4281. URL: https://link.aps.org/doi/10.1103/PhysRevLett.76.4281.

[6] Raymond Laflamme et al. “Perfect Quantum Error Correcting Code”. In: *Phys. Rev. Lett.* 77 (1 July 1996), pp. 198–201. DOI: 10.1103/PhysRevLett.77.198. URL: https://link.aps.org/doi/10.1103/PhysRevLett.77.198.

[7] A. M. Steane. “Error Correcting Codes in Quantum Theory”. In: *Phys. Rev. Lett.* 77 (6 June 1996), pp. 793–797. DOI: 10.1103/PhysRevLett.77.793. URL: https://link.aps.org/doi/10.1103/PhysRevLett.77.793.

[8] Peter W. Shor. “Scheme for reducing decoherence in quantum computer memory”. In: *Phys. Rev. A* 52 (4 Oct. 1995), R2493–R2496. DOI: 10.1103/PhysRevA.52.R2493. URL: https://link.aps.org/doi/10.1103/PhysRevA.52.R2493.

[9] Runyao Duan et al. “Multi-error-correcting amplitude damping codes”. In: *2010 IEEE International Symposium on Information Theory*. IEEE. 2010, pp. 2672–2676.

[10] David K Tuckett et al. "Tailoring surface codes for highly biased noise". In: *Physical Review X* 9.4 (2019), p. 041031.

[11] Peter Brooks and John Preskill. "Fault-tolerant quantum computation with asymmetric Bacon-Shor codes". In: *Physical Review A* 87.3 (2013), p. 032310.

[12] Chris Fisher. *IBM: Quantum Computing*. Apr. 2009. URL: `https://www.ibm.com/quantum-computing/`.

[13] Noson S Yanofsky and Mirco A Mannucci. *Quantum computing for computer scientists*. Cambridge University Press, 2008.

[14] Lov K Grover. "A fast quantum mechanical algorithm for database search". In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 212–219.

[15] Charles H Bennett et al. "Strengths and weaknesses of quantum computing". In: *SIAM journal on Computing* 26.5 (1997), pp. 1510–1523.

[16] Aamir Mandviwalla, Keita Ohshiro, and Bo Ji. "Implementing Grover's Algorithm on the IBM Quantum Computers". In: *2018 IEEE International Conference on Big Data (Big Data)*. 2018, pp. 2531–2537. DOI: `10.1109/BigData.2018.8622457`.

[17] Yulun Wang and Predrag S Krstic. "Prospect of using Grover's search in the noisy-intermediate-scale quantum-computer era". In: *Physical Review A* 102.4 (2020), p. 042609.

[18] Robert Wille, Rod Van Meter, and Yehuda Naveh. "IBM's Qiskit Tool Chain: Working with and Developing for Real Quantum Computers". In: *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*. 2019, pp. 1234–1240. DOI: `10.23919/DATE.2019.8715261`.

[19] accessed 2022-03-28. URL: `https://quantum-computing.ibm.com/`.

[20] accessed 2022-03-28. URL: `https://qiskit.org/documentation/apidoc/aer_noise.html`.

[21] accessed 2022-05-18. URL: `https://quantum-computing.ibm.com/services?services=systems&system=ibmq_lima`.

Appendix A

Source code

```
import math

from qiskit import IBMQ, Aer, transpile, assemble
from qiskit import execute
from qiskit.circuit import QuantumCircuit
import qiskit.providers.aer.noise as noise
from qiskit.visualization import plot_histogram
from qiskit.circuit.library import MCMTVChain
import csv


# Apply H-gates to put qubits in superstate
def initialize_s(qc, qubits):
    for q in qubits:
        qc.h(q)
    return qc

# Construct a diffuser for circuit of n qubits
def diffuser(nqubits):
    qc = QuantumCircuit(nqubits)
    for qubit in range(nqubits):
        qc.h(qubit)
    for qubit in range(nqubits):
        qc.x(qubit)
    qc.h(nqubits-1)
    qc.mct(list(range(nqubits-1)), nqubits-1)
    qc.h(nqubits-1)
    for qubit in range(nqubits):
        qc.x(qubit)
    for qubit in range(nqubits):
        qc.h(qubit)
    U_s = qc.to_gate()
    U_s.name = "Diffuser"
    return U_s


def execute_circ(n, noise_model=None, basis_gates=None, coupling_map=None):
    qs = list(range(n))
    grover_circuit = QuantumCircuit(n)
    grover_circuit = initialize_s(grover_circuit, qs)
    grover_circuit.append(diffuser(n), qs)
    grover_circuit.measure_all()
    grover_circuit.draw()

    n_seq = round(math.sqrt(math.pow(2,n)))
    result = execute(grover_circuit, Aer.get_backend(’qasm_simulator’),
                     noise_model=noise_model,
                     coupling_map=coupling_map,
                     basis_gates=basis_gates,shots=100000).result()
    counts = result.get_counts()
    return counts

def custom_noise(prob, gates, error_function):
    custom_noise = noise.NoiseModel()
    e = error_function(prob, 1)
    custom_noise.add_all_qubit_quantum_error(e, gates)
    custom_basis = custom_noise.basis_gates
    return custom_noise, custom_basis

# Build noise model from backend properties
provider = IBMQ.load_account()
backend = provider.get_backend(’ibmq_lima’)

real_noise = noise.NoiseModel.from_backend(backend)
real_basis = real_noise.basis_gates
coupling_map = backend.configuration().coupling_map

# build custom noise models
std_err = 3e-3
depolar_h, depolar_basis_h = custom_noise(std_err, ’h’, noise.depolarizing_error)
```

```
depolar_x, depolar_basis_x = custom_noise(std_err, ’x’, noise.depolarizing_error)
ampdamp_h, ampdamp_basis_h = custom_noise(std_err, ’h’, noise.amplitude_damping_er
ampdamp_x, ampdamp_basis_x = custom_noise(std_err, ’x’, noise.amplitude_damping_er
phasedamp_h, phasedamp_basis_h = custom_noise(std_err, ’h’, noise.phase_damping_erro
phasedamp_x, phasedamp_basis_x = custom_noise(std_err, ’x’, noise.phase_damping_erro

bitcount = [2,3,5]
for x in bitcount:
    clean = execute_circ(x)
    real = execute_circ(x, real_noise, real_basis, coupling_map)
    depolar = execute_circ(x, depolar_h, depolar_basis_h)
    ampdamp = execute_circ(x, ampdamp_h, ampdamp_basis_h)
    phasedamp = execute_circ(x, phasedamp_h, phasedamp_basis_h)

    with open(’test_’+ str(x)+ ’.csv’, ’w’) as f:
        f.write("Qubit,␣Baseline,␣Real,␣Depolarizing,␣Amplification,␣Phase\n")
        for key in clean.keys():
            f.write("%s,␣%s,␣%s,␣%s,␣%s,␣%s\n" % (key, clean[key], real[key],depolar[key],am

    legend = [’Without␣noise’, ’With␣noise’, ’Depolarizing’, ’Amplification␣dampening’, ’Pha
    plot_histogram([clean, real, depolar, ampdamp, phasedamp],
                   title=str(x) + "␣qubits",
                   legend=legend,
                   filename=f"{x}qubits.jpg",
                   figsize=(35,17))

    print(f"{x}␣qubits,␣clean:␣{clean}")
    print(f"{x}␣qubits,␣real:␣{real}")
    print(f"{x}␣qubits,␣depolarizing:␣{depolar}")
    print(f"{x}␣qubits,␣amplification␣dampening:␣{ampdamp}")
    print(f"{x}␣qubits,␣phase␣dampening:␣{phasedamp}")
```

Appendix B

IBMQ-lima parameters

Qubit	T1 (us)	T2 (us)	Frequency (GHz)	Anharmonicity (GHz)
Q0	154.35	202.87	5.03	-0.33574
Q1	114.9	132.35	5.128	-0.31835
Q2	126.8	11.49	5.247	-0.3336
Q3	142.81	106.94	5.303	-0.33124
Q4	20.67	24.59	5.092	-0.33447

Qubit	Readout assignment error	Single-qubit Pauli-X error
Q0	2.750e-2	2.028e-4
Q1	1.510e-2	1.647e-4
Q2	2.940e-2	2.823e-3
Q3	2.580e-2	2.573e-4
Q4	5.600e-2	8.025e-4

Table B.1: IBMQ-lima parameters, non-exhaustive [21]

.

TRITA – XXX-XXX 20XX:XX