KTH ROYAL INSTITUTE
OF TECHNOLOGY

DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING

FIRST CYCLE, 15 CREDITS

Impact of Noise on the Accuracy of Amplitude Amplification on the IBM Q Quantum Computer

EDVARD ALL

DILVAN SABIR

Stockholm, Sweden 2022

Impact of Noise on the Accuracy of Amplitude Amplification on the IBM Q Quantum Computer

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: Brus och dess påverkan på noggrannheten i amplitudamplifikation på kvantdatorn IBM Q

Abstract

Quantum computers promise immense speed up compared to even the most advanced supercomputers of today, but environmental noise introduces errors in the former, constituting a significant hurdle for realizing these promises. Quantum error-correcting codes have been suggested as a strategy to alleviate this hurdle, including codes optimized toward 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 design and construction of these optimized quantum error-correcting codes.

We simulate amplitude amplification, a central operation in an important quantum algorithm, Grover's algorithm, under different models of noise. Our results are inconclusive concerning discernment of the difference between different models 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

Kvantdatorer utlovar enorma prestandavinster till och med jämfört med dagens mest avancerade superdatorer. Tyvärr utgör miljöbrus ett betydande hinder i vägen mot förverkligandet av dessa löften. Kvantfelkorrigerande koder har föreslagits som en strategi för att motverka och minimera dessa hinder, till och med koder optimerade för beräkning under specifika brusmodeller. Att undersöka vilken inverkan olika typer av kvantfel har på resultaten av kvantberäkning kan därför vara instruerande för utformningen av dessa optimerade kvantfelskorrigerande koder.

Vi simulerar amplitudförstärkning, en central operation i en viktig kvantalgoritm för sökning i en osorterad mängd, under olika brusförhållanden. Våra resultat är ofullständiga med avseende på att urskilja skillnaden mellan olika typers brus effekt på beräkningsresultat, men är i linje med tidigare forskning i det avseendet att kvantfel redan vid fem qubit-kretsar påverkar resultaten så pass mycket att beräkningarna blir obrukbara.

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, introduce 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 realized. To efficiently make use of quantum error-correcting codes, it is necessary to understand how models of noise map to errors in quantum computation. Therefor this project aims to investigate the behavior of a prominent quantum algorithm, Grover's search algorithm, as subject to various models of noise. To this aim, we investigate the following question:

- How do the results of Grover's algorithm vary under different models of noise?

1.2 Scope and constraints

We aim to achieve the above goal 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 these machines for quantum computation (and mentioned noise models). Specifically, we will use up to five qubits, simulating one such machine, the IBMQ-lima.

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 the number of qubits in the circuit, as well as the channels of noise affecting the qubits. Because we simulate circuits of different numbers of qubits, and as oracle functions may be constructed in a number of different ways, we omit the oracle function (i.e. phase shift operation of Grover's) as it could obfuscate error analysis.

1.3 Thesis overview

This thesis is divided into four main parts. These are as follows:

Background: We provide a brief overview of the foundations of quantum informatics and noise in related computation, before explaining the workings of Grover's algorithm, and finally how to derive the gates from which we may construct the related amplitude amplification primitive circuit.

Method: We outline our simulational experiments and how we analyze the corresponding results.

Result: We present various comparisons of the results of amplitude amplification under different variants of noise.

Discussion: We attempt to summarize and discuss our results, highlighting various limitations of our work as well as future avenues of research.

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 analogous to the classical bit, except 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 the qubit to be in a *superposition* of both states simultaneously. This means that a qubit does not have a discrete value, but rather it is in a state denoting its probability of collapsing to a specific value 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 $|0\rangle$ (pronounced ket), $|1\rangle$ 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 canceling each other out (much like two waves colliding may cancel each other).

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.

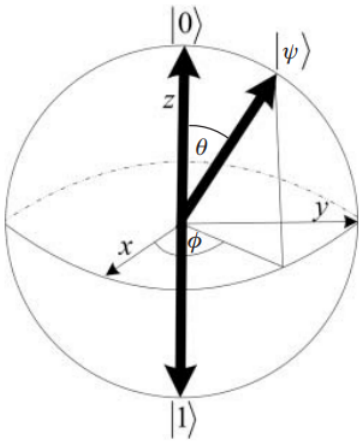


Figure 2.1: A geometric representation of a qubit, the Bloch sphere [13, p. 161].

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. For example, take the set of Pauli-gates (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 and Z flip the Bloch sphere halfway around the corresponding axis. Another operation of note is the Hadamard, which may be used to place a qubit in a superposition:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

2.3 Noise in quantum computing

Real-world quantum devices are subject to environmental noise and therefore 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, increasing 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. However, this strategy might not be sufficient as it may well be the case that gates cannot satisfy this criterion.

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, give 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 a 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.

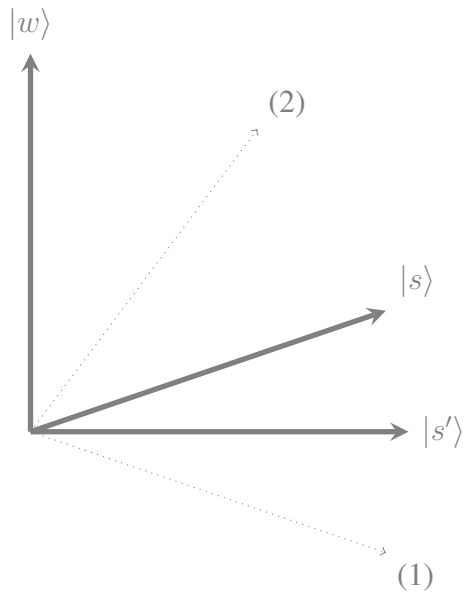


Figure 2.2: Geometric intuition for Grover's algorithm. The amplitude of the desired state $|w\rangle$ is amplified by: (1) reflecting $|s\rangle$ over $|s'\rangle$, and (2) reflecting back over $|s\rangle$.

Before overviewing the algorithm itself, we outline a geometric intuition for how and why Grover's algorithm works. In figure 2.2 we represent the initial superposition of the system by some vector $|s\rangle$, which we wish to transform into the desired (winning) state $|w\rangle$. To achieve this, we imagine the state $|s'\rangle$ orthogonal to $|w\rangle$, and reflect our state around this $|s'\rangle$. We then reflect this new state around $|s\rangle$, and the amplitude of the system is now closer to $|w\rangle$ than before our operation. This process is referred to as amplitude amplification.

Grover's algorithm makes use of this very method, 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 $|s\rangle$.
3. Repeat $\sqrt{2^n}$ times:
   (a) Using the oracle function f, flip the phase of the desired state(s), which corresponds to (1) in figure 2.2.
   (b) Invert phase of the state about the mean, which corresponds to (2) in figure 2.2.

4. Measure the system.

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

2.4.1 Constructing the amplitude amplification primitive

To invert the phase of the quantum system about its mean, we first find this mean, which may be achieved with a matrix all elements of which equaling $\frac{1}{2^n}$, where n is the number of qubits:

$$A = \begin{bmatrix} \frac{1}{2^n} & \cdots & \frac{1}{2^n} \\ \vdots & \ddots & \vdots \\ \frac{1}{2^n} & \cdots & \frac{1}{2^n} \end{bmatrix},$$

Then by multiplying our state $|s\rangle$ by $2A - \mathbf{I}$, $\mathbf{I}$ denoting the identity matrix, we may achieve this inversion about the mean. We introduce the complex conjugate of some state $|\psi\rangle$, $\langle\psi|$ (pronounced bra). We are now ready to decompose this operation into a circuit of gates.

$$\begin{aligned} (2A - \mathbf{I})\,|s\rangle &= 2\,|s\rangle\,\langle s| - \mathbf{I} \\ &= H^{\otimes n}(2\,|0\rangle\,\langle 0| - \mathbf{I})H^{\otimes n} \\ &= H^{\otimes n}(X^{\otimes n} \times CZ \times X^{\otimes n})H^{\otimes n} \end{aligned}$$

where CZ denotes the multi-controlled Z gate. Note that $Z = HXH$, a fact which is used in construction of the circuits (which may be viewed in figure 3.1 on page 12).

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).

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 behavior 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 amplitude amplification primitives 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 Noise profiles

The following noise profiles were introduced to the simulated systems (parenthesis denote introduction of labels used in later figures and tables):

- No noise (Baseline)
  - Represents the circuits as is with no noise models.
- Realistic noise (Real)
  - Represents the circuits with a noise model generated based upon actual IBM Quantum computers.
- Depolarizing errors only (Depolarizing)
  - This error model can be interpreted as a uniform contraction of the Bloch sphere, corresponding to a loss of quantum information.
- Amplitude damping errors (Amplitude)
  - May be interpreted as a shrinkage of the Bloch sphere towards the north pole, and so an amplified amplitude of $|0\rangle$.
- Phase Damping errors (Phase)
  - May be interpreted as a contraction of the Bloch sphere in the XY-plane, leaving Z as was.

The no-noise variant is then the established baseline profile, which we expect to be a uniform distribution as the amplitude amplification primitive 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.1) in order to ensure that simulated conditions were reasonably realistic: a rough mean of the qubit error was applied to all qubits. No readout error was applied, however.

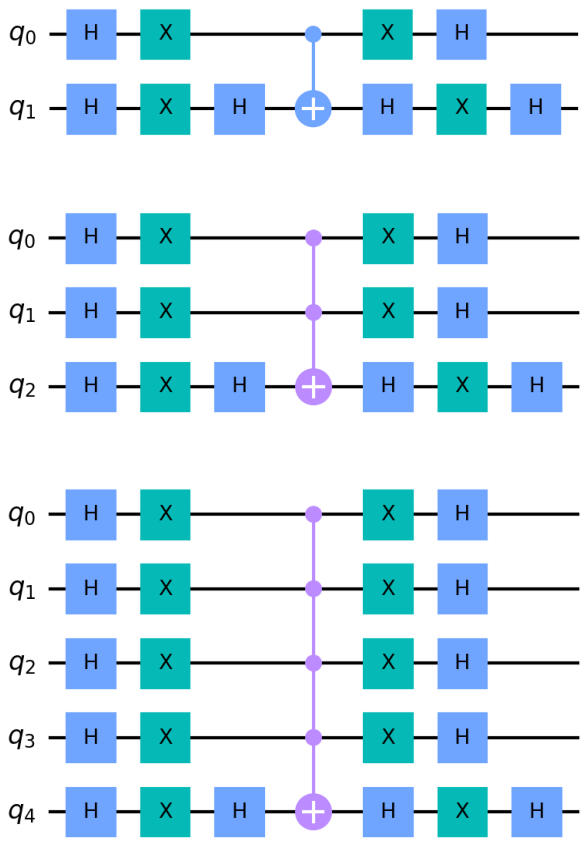


Figure 3.1: Circuits for amplitude amplification primitive of size two, three, and five qubits respectively.

3.3 Evaluation

To evaluate the resulting frequencies, we first calculated the sum squared error (SSE) of each respective noise profile relative to the baseline profile as follows:

$$SSE = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

where $n = 100,000$ (the number of times a circuit was run), Y is the mean noiseless baseline frequencies, and $\hat{Y}$ mean of one of the noise profiles. A value closer to 0 indicates that the impact of noise is low and the profile operates close to the baseline, whereas higher values indicate a high impact of noise on the results. Notably, one source of significant error will contribute more to the SSE than multiple sources of minor errors (due to the squaring of the inner term), which is just the property we seek; many minor errors will be less likely to degenerate the results of the complete Grover's algorithm (i.e. oracle included).

By the law of large numbers, the errors should follow a normal distribution, and so to further evaluate these errors, we conducted an SSE-related test, the one-tailed two-sample F-tests of the variance ratio, between noise and baseline.

Chapter 4

Results

To visualize the results, we provide heatmaps of measured errors in figures 4.1 and 4.2, represented by the absolute difference between the mean frequency of a measured state under each respective noise model and the baseline. SSE was calculated for each respective profile and circuit and is available in table 4.4. To visualize these errors in absolute terms, histograms over states and their precise mean frequency is provided in figure 4.3, 4.4 and 4.5.

	Real	Depolarizing	Amplitude	Phase
F	32.965148	0.80363634	0.905394	2.144929
P(F≤f) one-tail	0.0085	0.430827	0.468404	0.273461
F Critical one-tail	9.276628	0.10798	0.10798	9.276628

Table 4.1: F-test results for 2 qubits relative to the baseline profile

	Real	Depolarizing	Amplitude	Phase
F	318.223702	1.295730	2.324436	3.764796
P(F≤f) one-tail	3.18516E-08	0.370589	0.144112	0.050709
F Critical one-tail	3.787044	3.787044	3.787044	3.787044

Table 4.2: F-test results for 3 qubits relative to the baseline profile

Results of the conducted F-tests are given in table 4.1, 4.2 and 4.3. A significant difference in variance can be observed for the realistic noise profile on two, three, and five qubits. Ditto can also be observed for the amplitude damping error model on five qubits, although this difference is quite marginal.

	Real	Depolarizing	Amplitude	Phase
F	93.547470	1.606256	2.061986	1.123137
P(F≤f) one-tail	3.14934E-23	0.096337	0.023997	0.374259
F Critical one-tail	1.822132	1.822132	1.822132	1.822132

Table 4.3: F-test results for 5 qubits relative to the baseline profile

	2 Qubits	3 Qubits	5 Qubits
SSE Real	3.26600E-02	1.32026E-03	6.28416E-02
SSE Depolarizing	6.50000E-03	1.75190E-05	1.96362E-03
SSE Amplitude	9.44000E-03	1.54174E-05	1.81844E-03
SSE Phase	5.84000E-03	1.79882E-05	1.25050E-03

Table 4.4: SSE of the different profiles relative to the baseline profile.

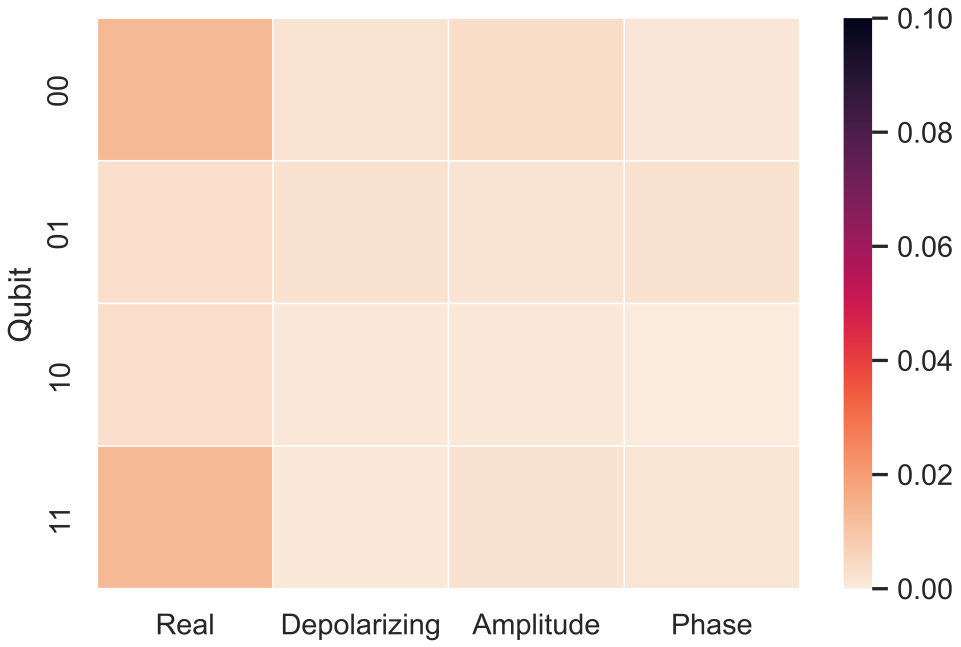


Figure 4.1: Absolute difference of mean frequency of the measured 2 qubit circuit states under noisy and noiseless profile, after 100,000 runs.

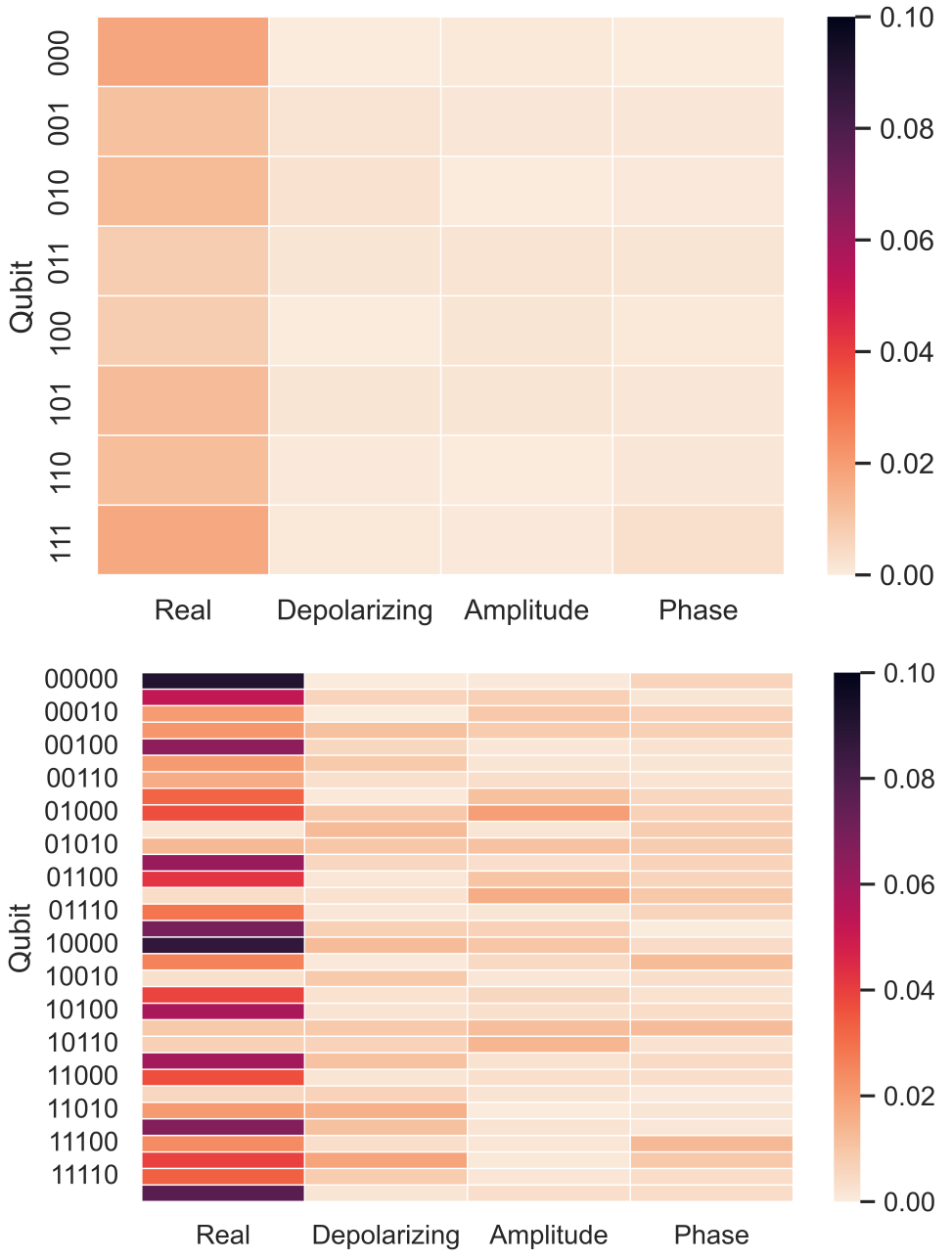
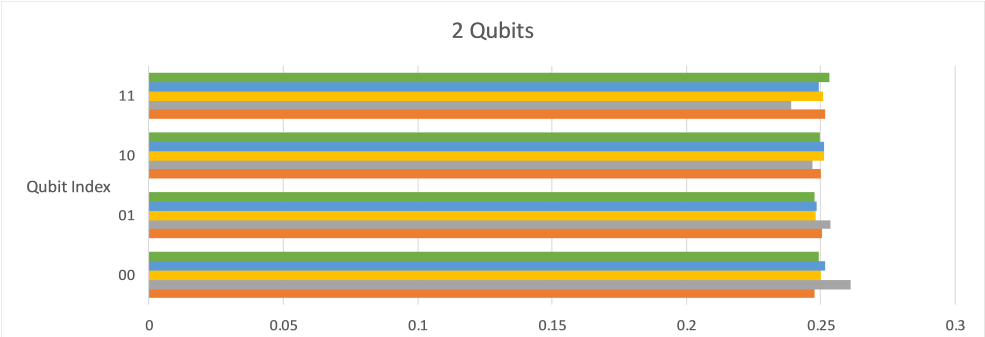
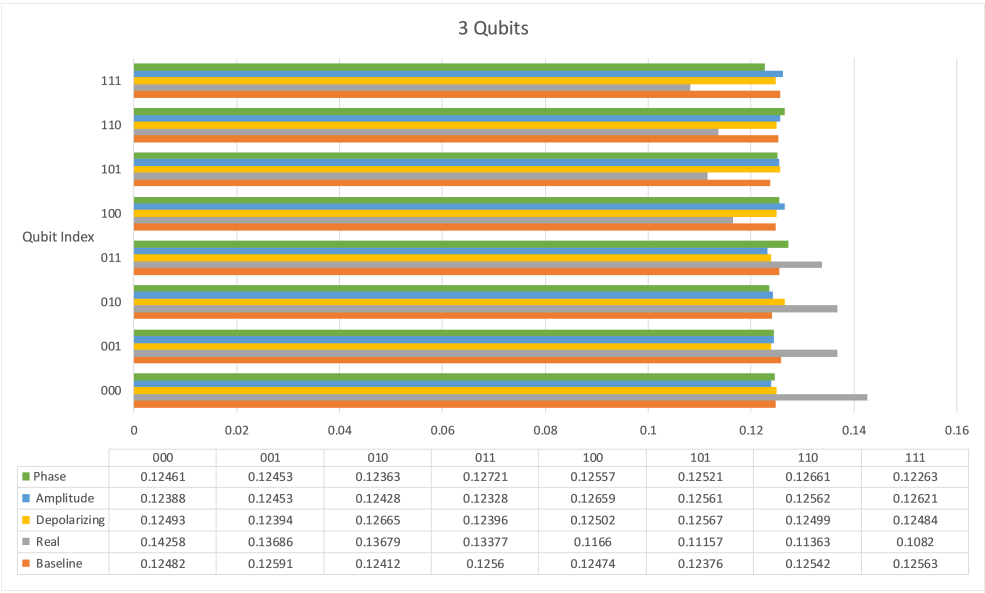


Figure 4.2: Absolute difference of mean frequency of the measured 3 and 5 qubit circuit states under noisy and noiseless profile, after 100,000 runs.



	00	01	10	11
Phase	0.24913	0.24771	0.2498	0.25336
Amplitude	0.25144	0.24831	0.25121	0.24904
Depolarizing	0.24999	0.24793	0.25119	0.25089
Real	0.2609	0.25353	0.24672	0.23885
Baseline	0.2478	0.2503	0.25013	0.25177

Figure 4.3: Histogram of the distribution of states for the 2 qubit circuit after 100,000 runs.



	000	001	010	011	100	101	110	111
Phase	0.12461	0.12453	0.12363	0.12721	0.12557	0.12521	0.12661	0.12263
Amplitude	0.12388	0.12453	0.12428	0.12328	0.12659	0.12561	0.12562	0.12621
Depolarizing	0.12493	0.12394	0.12665	0.12396	0.12502	0.12567	0.12499	0.12484
Real	0.14258	0.13686	0.13679	0.13377	0.1166	0.11157	0.11363	0.1082
Baseline	0.12482	0.12591	0.12412	0.1256	0.12474	0.12376	0.12542	0.12563

Figure 4.4: Histogram of the distribution of states for the 3 qubit circuit after 100,000 runs.

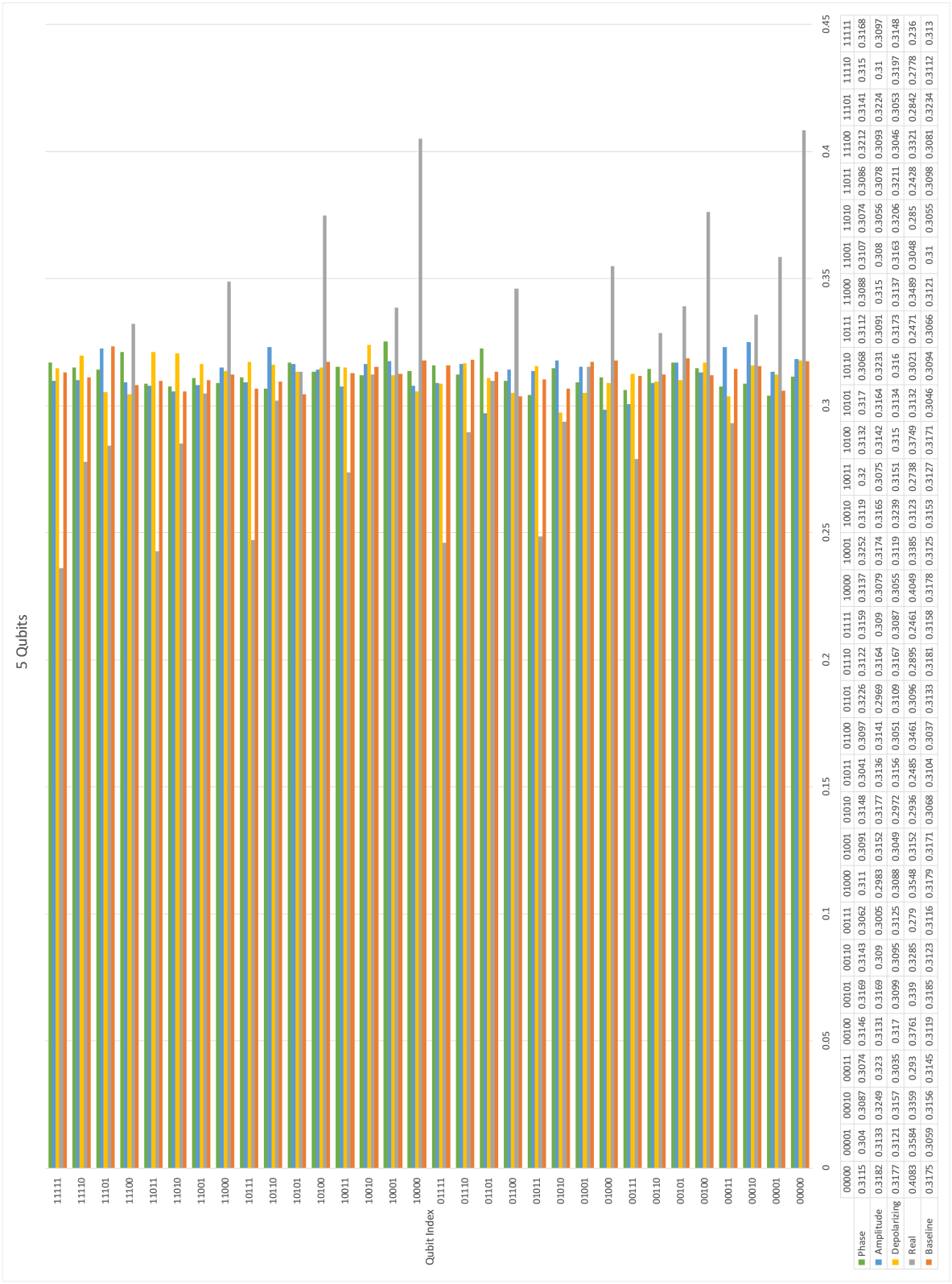


Figure 4.5: Histogram of the distribution of states for the 5 qubit circuit after 100,000 runs.

Chapter 5

Discussion

As can be plainly observed already in comparing figure 4.3 and figure 4.4, the variance of the resulting distributions increases dramatically as the number of qubits in the circuits increase. At five qubits, the distribution under realistic noise degenerates completely from the desired uniformity, which likely renders the results unusable for the process of amplitude amplification. This is demonstrated in real terms by the differences in SSE, as the SSE for the five qubit circuit under realistic noise is roughly double that of the two-qubit ditto.

Comparing SSEs of the different noise models suggests that the type of error matters little regarding errors of the amplitude amplification primitive. This is in line with conducted F-tests, which found a significant difference in the variance of error distributions under the realistic noise profiles only (exception being the amplitude damping error profile on five qubits, although the difference there is marginal).

However, there are a number of limitations to our method that makes it difficult to draw clear conclusions from the results. In the interest of transparency and ease of implementation, we opted for simplistic custom noise profiles, 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 an oracle-less Grover operation, thus simplifying the circuit and likely lowering the accumulated number of errors of the output state.

There are additional limitations to our analysis of the errors. Firstly, comparing SSEs over circuits of different sizes could be misleading, as the residual differences will spread over a greater number of events. This means that although real errors are relatively large at five qubits, the corresponding SSE is not much greater than that of the ditto two-qubit circuit in absolute terms. Sec-

ondly, it is difficult to interpret these results in regards to what constitutes the error threshold for the Grover operator. This was out of the scope of this study, but it could have been interesting to contrast our results against this threshold. Building upon this, it would also be interesting to investigate how error under this threshold impacts the convergence of the quantum system under amplitude amplification towards the desired amplitude, with regards to the number of evaluation steps required to reach an acceptable probability of measuring the correct state.

Therefore, further studies could improve on these noise models, as to ascertain conclusive results regarding the effect various noise has on resulting errors of the output. Additionally, oracles could be constructed and introduced to the circuit, whereby the errors of this circuit could be compared to the oracle-less one. Furthermore, more fine-grained methods of analysis could be adopted to cast a better light on errors in computation.

5.1 Conclusion

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

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 csv
import math

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


# 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_error)
ampdamp_x, ampdamp_basis_x = custom_noise(std_err, 'x', noise.
    amplitude_damping_error)
```

```
77  phasedamp_h, phasedamp_basis_h = custom_noise(std_err, 'h', noise.phase_damping_error
        )
78  phasedamp_x, phasedamp_basis_x = custom_noise(std_err, 'x', noise.phase_damping_error
        )
79
80  bitcount = [2, 3, 5]
81  for x in bitcount:
82      baseline = execute_circ(x)
83      real = execute_circ(x, real_noise, real_basis, coupling_map)
84      depolarizing = execute_circ(x, depolar_h, depolar_basis_h)
85      amplitude = execute_circ(x, ampdamp_h, ampdamp_basis_h)
86      phase = execute_circ(x, phasedamp_h, phasedamp_basis_h)
87
88      with open('test_' + str(x) + '.csv', 'w') as f:
89          f.write("Qubit, Baseline, Real, Depolarizing, Amplitude, Phase\n")
90          for key in clean.keys():
91              f.write("%s, %s, %s, %s, %s, %s\n" % (
92                  key, baseline[key], real[key], depolarizing[key], amplitude[key], phase[key]))
93
94      legend = ['Without noise', 'With noise', 'Depolarizing', 'Amplification dampening', '
          Phase dampening']
95      plot_histogram([baseline, real, depolarizing, amplitude, phase],
96                     title=str(x) + " qubits",
97                     legend=legend,
98                     filename=f"{x}qubits.jpg",
99                     figsize=(35, 17))
100
101     print(f"{x} qubits, baseline: {baseline}")
102     print(f"{x} qubits, real: {real}")
103     print(f"{x} qubits, depolarizing: {depolarizing}")
104     print(f"{x} qubits, amplification dampening: {amplitude}")
105     print(f"{x} qubits, phase dampening: {phase}")
```

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]

.