Exqaliber Stage 1 Report

James Cruise[1], Joseph Tedds[1], Camille de Valk[2], and Walden Killick[1]

[1]Cambridge Consultants, Cambridge, UK
[2]Capgemini QLab

TBD

1 Introduction

Quantum amplitude estimation (QAE) is a fundamental subroutine in many application areas including quantum chemistry, machine learning, finance, and more. It is the problem of, given access to an $(n+1)$-qubit oracle $\mathcal{A}$ such that $\mathcal{A}|0\rangle = \sqrt{a}|\phi_1\rangle|1\rangle + \sqrt{1-a}|\phi_0\rangle|0\rangle$, estimating a. This is a problem of great current interest due to its promise of a quantum quadratic speedup, including potentially on noisy near-term devices.

The first such algorithm, proposed by Brassard et al. [1], works by combining the quantum phase estimation (QPE) algorithm [2] with the Grover iteration operator. Although this achieves a provable quantum speedup, this approach is unsuitable for use on near-term devices primarily due to its large circuit depth stemming from (1) the quantum Fourier transform (QFT) [3] and (2) the controlled unitaries required to prepare the input state of the QFT.

[cdv 1] I don't think we should bring up the math in the first paragraph. Maybe we can keep it a bit more high-level.

Several recent approaches have studied algorithms achieving the same or similar asymptotic speedups without the use of the QFT, in particular by repeated sampling of the state after only repeated applications of the Grover operator [4, 5, 6]. The underlying motif of this paradigm is the learning of the amplitude by statistical sampling, which is in contrast to Brassard et al.'s method which allows for direct access to the amplitude with high probability. Each of these approaches has its own problems with regards to practical implementation on near-term devices; most notably, rapidly growing gate depths, large constant-factor overheads, and expensive controlled-Grover operations. All of these are problematic in the presence of noise as repeated applications of the Grover operator causes the state to decohohere; after which, without a noise model, information supposedly gained about the amplitude may be incorrect or skewed.

In this work, we present hybrid quantum-classical algorithms for QAE with a focus on minimal circuit depth and measurement counts, and taking into account the noisiness of near-term devices. (tbc. after we know what's in the rest of the report)

James Cruise: james.cruise@cambridgeconsultants.com
Joseph Tedds: joseph.tedds@cambridgeconsultants.com
Camille de Valk: camille.de.valk@capgemini.com
Walden Killick: walden.killick@cambridgeconsultants.com

1.1 Amplitude estimation

Problem 1 (Quantum Amplitude Estimation)**.** *Given some* $(n+1)$*-qubit oracle* $\mathcal{A}$ *which generates the state*

$$\mathcal{A}\left|0\right\rangle = \sqrt{a}\left|\phi_1\right\rangle\left|1\right\rangle + \sqrt{1-a}\left|\phi_0\right\rangle\left|0\right\rangle,$$

where $\left|\phi_1\right\rangle$ *and* $\left|\phi_0\right\rangle$ *are arbitrary normalised* n*-qubit states and* $0 \leq a \leq 1$*, quantum amplitude estimation (QAE) is the problem of estimating the unknown amplitude* $\sqrt{a}$*.*

The 'naive', classical way to solve this problem is to simply prepare and measure this state many (say N) times, counting the number of times $\left|\phi_1\right\rangle\left|1\right\rangle$ was observed (say K), then define the estimate for $\sqrt{a}$ as $\sqrt{K/N}$. Chebyshev's inequality tells us that taking $N \in O(\varepsilon^{-2})$ samples is sufficient to approximate $\sqrt{a}$ within additive error ε (with, say, 99% confidence). Furthermore, this is asymptotically optimal [7]. By contrast, in a manner which will be made more precise in following sections, quantum computers allow us to modify the state before measurement, which allows us to achieve the same degree of accuracy using only $O(\varepsilon^{-1})$ queries (known as the Heisenberg limit), achieving a quadratic quantum speedup [1].

An important and general application of QAE is Monte Carlo estimation [8, 9, 5] - a method for estimating the mean value of a function via random sampling. More precisely, given some function $f : \{0,1\}^n \to [0,1]$, the task is to estimate the mean value

$$\mathbb{E}[f(X)] = \frac{1}{2^n}\sum_{x=0}^{2^n-1} f(x).$$

Montanaro [9] presents a method for encoding this value in the amplitude of a quantum state, thereby reducing this problem to a case of amplitude estimation, which in turn offers a quadratic speedup for Monte Carlo estimation over classical methods.

[cdv 2] Monte Carlo is a way to estimate a general integral, right? Not just the mean.

1.1.1 Statistical amplitude estimation

Although the original QAE algorithm based on QPE is still the superior algorithm to run on a fault-tolerant quantum computer, the noisiness of near-term devices makes all algorithms utilising the QFT, including the aforementioned algorithm, very difficult to implement (primarily due to the large circuit depth). Thus, the current paradigm in the design of QAE algorithms is to sample the state after amplitude amplification of the state $\mathcal{A}\left|0\right\rangle$, refining the estimate for $\sqrt{a}$ and increasing confidence in the estimate with consecutive samples.

By setting $\theta = \sin^{-1}\sqrt{a}$, we can rewrite the state as

$$\mathcal{A}\left|0\right\rangle = \sin\theta\left|\phi_1\right\rangle\left|1\right\rangle + \cos\theta\left|\phi_0\right\rangle\left|0\right\rangle.$$

Now consider the Grover iterate given by

$$\mathcal{U} = \mathcal{A}\left(2\left|0^{n+1}\right\rangle\!\left\langle 0^{n+1}\right| - I_{2^{n+1}}\right)\mathcal{A}^{-1}\left(I_{2^n}\otimes Z\right)$$

as in Figure 1. In the subspace spanned by $\left|\phi_1\right\rangle\left|1\right\rangle$ and $\left|\phi_0\right\rangle\left|0\right\rangle$, by a sequence of reflections, each application of $\mathcal{U}$ performs a rotation of angle 2θ towards $\left|\phi_1\right\rangle\left|1\right\rangle$. That is,

$$\mathcal{U}^m\mathcal{A}\left|0\right\rangle = \sin(2m+1)\theta\left|\phi_1\right\rangle\left|1\right\rangle + \cos(2m+1)\theta\left|\phi_0\right\rangle\left|0\right\rangle$$

and so $\mathcal{U}^m$ takes the probability of measuring $\left|\phi_1\right\rangle\left|1\right\rangle$ from $\sin^2\theta$ to $\sin^2(2m+1)\theta$.

[cdv 3] We should be consistent in using k or m. Later, Joe and I talk about k.

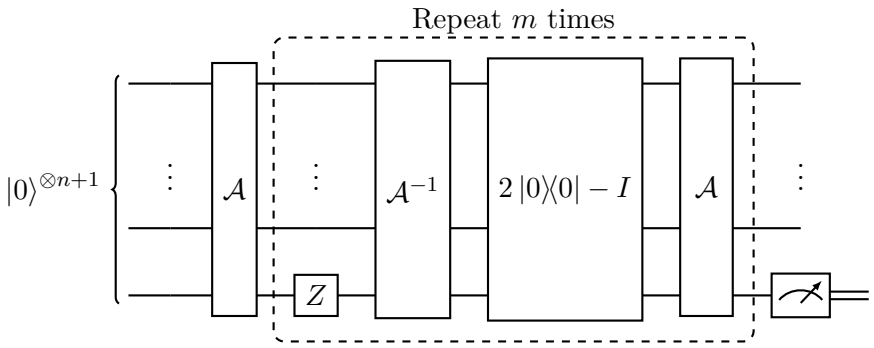


Figure 1: Circuit for amplitude amplification. The probabilities for the measurement outcomes are $\Pr(|1\rangle) = \sin^2(2m+1)\theta$ and $\Pr(|0\rangle) = \cos^2(2m+1)\theta$.

1.1.2 Decoherent noise model

Performing Bayesian inference without a noise model on a device which in reality is noisy may cause incorrect information to be attained about the angle θ. For example, if the true angle $\theta = 0$, then, if the system were truly noiseless, no number of Grover iterates should affect the state and make it possible to measure $|\phi_1\rangle|1\rangle$. However, in the realistic noisy case, depolarisation stemming from the application of gates may cause the probability of measuring $|\phi_1\rangle|1\rangle$ to rise above 0 nonetheless. If we measure this state, without a noise model, our understanding of the angle effectively eliminates the case $\theta = 0$. The presence of a noise model mitigates this issue by effectively decreasing confidence in the measurement outcome as the number of Grover iterations increases, thereby retaining the possibility that the non-zero probability was the result of noise.

In the presence of depolarising noise, it is reasonable to assume that consecutive applications of $\mathcal{A}$ and its inverse cause, on average, the probability of measuring $|\phi_1\rangle|1\rangle$ to tend to $\frac{1}{2}$, at which point no information about the original state is recoverable.

In the ideal (fault-tolerant) model, the probability of measuring $|\phi_1\rangle|1\rangle$ after m iterations of U to $\mathcal{A}|0\rangle$ is $\sin^2(2m+1)\theta$, which can be rewritten as

$$\sin^2(2m+1)\theta = \frac{1}{2}\left(1 - \cos(4m+2)\theta\right).$$

If we assume that each iteration of $\mathcal{A}$ or its inverse dampens the oscillating term (which contains information about θ) by some factor $e^{-\lambda}$, where $\lambda \geq 0$ is a constant corresponding to the noisiness of the system, then we can model the new probability of measuring $|\phi_1\rangle|1\rangle$ after m iterations of $\mathcal{U}$ as

[cdv 4] Here we have the 4 again, instead of the 2.

$$\Pr(|\phi_1\rangle|1\rangle) = \frac{1}{2}\left(1 - e^{-\lambda(2m+1)}\cos(4m+2)\theta\right).$$

Here, the exponent contains the term $2m+1$ since we need one application of $\mathcal{A}$ to prepare the state, then each iteration of $\mathcal{U}$ requires one call to $\mathcal{A}$ and another to its inverse.

(maybe write some more about what we do with this noise model - tbc. after the rest of the report is written)

1.2 Related work

Brassard et al. [1] were the authors of the first QAE algorithm, now often referred to as canonical QAE. The idea is that the Grover operator $\mathcal{U}$ as defined above contains the

amplitude of the corresponding state in its eigenvalues, and thus the QPE algorithm can be used with $\mathcal{U}$ to extract this value. This method allows for direct access to the amplitude to within additive error ε with high probability using $O(\varepsilon^{-1})$ queries to $\mathcal{A}$, which makes it still the superior algorithm for QAE on a fault-tolerance quantum computer. However, its reliance on controlled Grover iterates and the QFT make it very difficult to implement on near-term devices, which has inspired much subsequent work on QAE without the use of the QFT.

Suzuki et al. [5] propose a QFT-free QAE algorithm based on maximum likelihood estimation. Although lacking a rigorous proof, numerical simulation seems to show that their algorithm achieves the asymptotically optimal scaling of measurements. However, the scheme which achieves this scaling incurs an exponentially increasing number of Grover iterates at each step, which is problematic especially in the presence of noise. Furthermore, the choice of number of Grover iterates is not dependent on previous measurement outcomes, which suggests that there is further optimisation to be achieved in minimising the circuit depth.

Wie [6] sketches another QFT-free QAE algorithm based on Hadamard tests, similar to that which is used in iterative phase estimation as presented by Kitaev [2]. Apart from also lacking a proof of optimality, Wie's algorithm uses the more expensive controlled-Grover operations.

Aaronson and Rall [4] also present an algorithm for QFT-free QAE, and were the first to rigorously prove a quadratic speedup for such an algorithm. Although this algorithm achieves the optimal asymptotic complexity, large constant-factor overheads make it impractical for use even in the fault-tolerant case [10].

Grinko et al. [10] present an algorithm called iterative QAE which combines ideas from previous works and greedily chooses the number of Grover iterates at each step to maximise the quantum Fisher information. They prove that their algorithm is optimal up to a double-logarithmic factor and has much smaller constant factors than comparable algorithms.

Giurgica-Tiron et al. [11] design algorithms which interpolate between classical and quantum amplitude estimation algorithms with the aim of utilising parallelism to minimise overall circuit depth. Their algorithms achieve a query complexity of $\tilde{O}\left(\varepsilon^{-(1+\beta)}\right)$ where $\beta \in (0, 1]$ is some parameter corresponding to the balance between classical and quantum queries.

Smith et al. [12] present an algorithm for QFT-free QPE (which can equivalently be used for QAE by replacing the unitary operator with the Grover iterate) which is also based on maximising Fisher information. They prove that it achieves the Heisenberg limit in the noiseless case, and prove that it achieves the best possible query complexity in the presence of depolarising noise.

1.2.1 Quantum phase estimation

tbc. how much should we write about this? already discussed it a bit in previous sections

The first algorithm for QAE, presented by Brassard et al. [1], achieves its objective by applying the QPE algorithm [2] to the Grover iterate defined above.

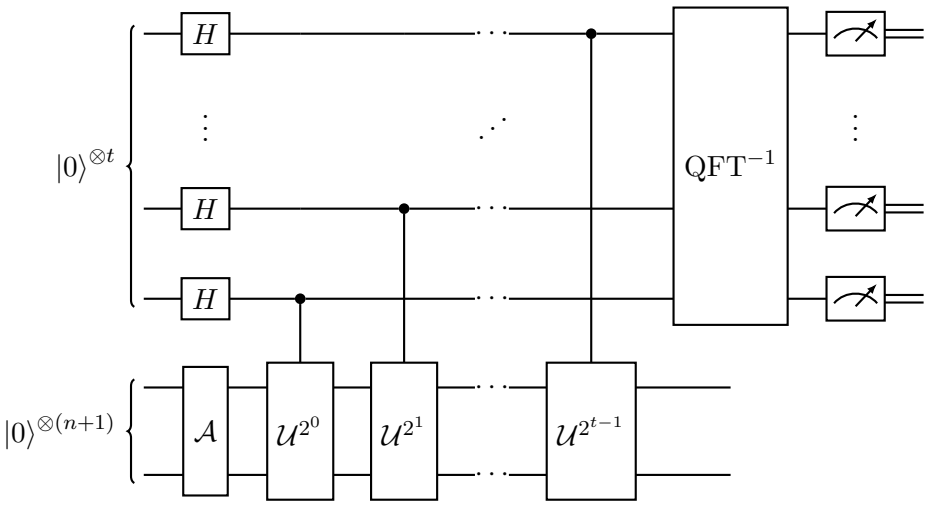


Figure 2: Amplitude estimation by phase estimation

1.3 Contribution

(to be written once we know what's in the rest of the paper)

2 Algorithm

As discussed in **??** the algorithm is made up of two parts, the sampling algorithms which provides a dynamic sampling scheme to reach the desired accuracy and the post-processing algorithm which generate the final estimate of θ in an offline fashion. The reason for this two stage approach is that maintaining a true representation of the current state and uncertainty during the sampling scheme is too computationally intensive so we utilise a low dimensional approximation. In most cases provides a high quality estimate which enables us to carry out an online optimisation of sample number and depth. Once all the samples are obtain rather than return the approximation, which does not have the full predictive power, we re-estimate θ utilising the full data set that we have obtained during the sampling. This means the errors introduced by the approximation procedure do not effect the final estimate.

In this paper the novel work is in the sampling procedure and the associated modelling. For the post-processing we use the maximum likelihood estimate (MLE) originally described in [**?**]. For this reason here we concentrate on the sampling algorithm and include a short introduction to the MLE at the end of this section.

2.1 Sampling algorithm

Here we describe the sampling algorithm and associated theoretical underpinning initially in the noiseless case and then introduce the adjustment required to deal with the introduction of depolarising noise. The fundamental idea is that we maintain a current state representing our belief about the value of the parameter of interest, μ, and the current level of uncertainty, σ^2. Then at each step we decide if enough samples have been obtained, i.e. σ^2 is small enough for the required accuracy, or to generate another sample.

If another sample is generated we need to select the depth at which to sampling circuit is executed and using the result update the state, i.e. our belief. In both cases we use a Bayesian framework to do this, see 2.1.1. Within this framework, the uncertainty σ^2 is an estimate of the posterior variance for θ and μ, the current belief of θ, is an estimate of the posterior mean. At each step the next sampling depth is selected to minimise the expected posterior variance given the estimate of the current belief, i.e. the depth which in expectation moves us closest to terminating the sampling. After executing the sampling circuit, the state is updated by calculating the posterior mean and variance. This is where the main approximation is used, unfortunately maintaining and manipulating an estimate of the true posterior distribution is computationally intensive so after every step we approximate the posterior distribution by a normal distribution. This is then used as the prior distribution for the next step, for more details see Section 2.1.1.

Having described the algorithm we now provide a pseudocode specification of the algorithm, see 1. There are couple of points to note from this pseudocode, firstly the procedure 'Sample' executes the quantum circuit shown in figure **??** where d is the number of times the Grover operator is applied. The measurement of 0 or 1 is then stored as x. Secondly, as described in Section 2.2 we post process the obtained samples to produce the final estimate of θ so 'Store' stores the obtained sample with the associated depth for later use. Thirdly, there are three closed form mathematical functions, g_1, g_2, h, used within the pseudocode defined in Section 2.1.2. Note that the only change to this sampling algorithm when considering the depolarizing noise is to g_1, g_2, h otherwise the algorithm stays the same.

Algorithm 1 Pseudocode for sampling algorithm

Require: $(\mu, \sigma^2), \epsilon, \alpha$
  while $\sigma > \Phi^{-1}(1-\alpha)\epsilon$ **do** ▷ Φ is the normal distribution function
    $d \leftarrow \arg\max_{d>0}(h(\mu, \sigma^2, d))$ ▷ See equation 4
    $x \leftarrow \textsc{Sample}(\mathrm{d})$ ▷ Execute circuit depth at d
    $\mu \leftarrow g_1(\mu, \sigma^2, x, d)$
    $\sigma^2 \leftarrow g_2(\mu, \sigma^2, x, d)$ ▷ Update state, see equations 2 and 3
    $\textsc{Store}((x, d))$ ▷ Store observation and sample depth for post-processing
  end while

2.1.1 Bayesian framework

Through the design of this algorithm we have embraced a Bayesian approach [] and used this to generate a sequential sampling algorithm. Note, an important difference from the standard sequential algorithms is that normally we are only interested in deciding when to stop sampling but here we also have control of the sampling distribution.

The fundamental feature of this approach is that at all stages we maintain a belief about our current knowledge of the unknown parameter, θ, through a probability distribution, $\pi(\theta)$. This belief is updated after each sample using Bayes formula [] and the current belief is used to make the decision of whether to stop or generate a new sample and in that case what depth to sample at.

Unfortunately, keeping the full probability distribution is computationally infeasible so instead we approximate our belief by a normal distribution, i.e. $\pi(\theta)$ $N(\mu, \sigma^2)$, which is fully described by two parameters the mean, μ, and the variance, σ^2. Therefore at each stage we approximate the posterior distribution, which becomes the prior when considering the next sample, by a normal distribution with the same variance and mean as the one step posterior distribution. Note that this mean and variance will not necessarily match the posterior mean and variance if all the obtained samples where considered due to the repeated approximation.

The normal distribution is a sensible approximation since under suitable conditions, see [], in the limit of a large number of samples the posterior limits to a normal distribution. Further to this using the normal distribution as our prior belief, we are able to use Bayes theorem to produce analytic expressions for a number of key posterior parameters including the posterior mean and variance and expected posterior variance, see the next section. This is important because we could obtain estimates of these through sampling schemes [] but as we will discuss in Section 2.1.5 this leads to a systematic bias.

Though using a normal approximation will generally provide a good ongoing estimate but there are approximation errors which can be substantial in nature. For some experimental examples of this see Section**??**. This is the reason that we carry out a two stage algorithm and post process the samples to provide the final estimate for the parameter of interest, Section 2.1.2. Two relevant examples of approximation errors are:

- The parameter we are interested θ is best thought as angle, i.e. 0 and 2π should be identified with each other. This means that we have to be especially careful near boundary points, for example 0.
- We are sampling from a Bernoulli distribution where the probability of a 1 is determined by the unknown parameter θ and the circuit depth. Unfortunately, as

discussed in Section **??** the exact mapping to probability has a degeneracy when consider depths greater than 1, i.e. there are multiple values of θ which give the same value of a 1. If care is not taken this can lead to a multimodal posterior distribution.

2.1.2 Posterior properties

Two keys steps in the algorithm is calculating for a single sample the expected posterior variance, $Var(\theta)$ which is used to decide on the next sample depth and the posterior mean and variance, $\mathbb{E}(\theta|X = x)$ and $Var(\theta|X = x)$. As we have already seen in the pseudocode these play a significant role in our algorithm. We are able to calculate all three and obtained a closed form expression under the assumption that the prior follows a normal distribution with known mean and variance. Our derivation follows closely the work described in [] which considers a class of likelihoods of the same form as we see in this work. Details of the derivation can be found in Appendix **??** and here we only include the final formulas for use with the algorithm.

$$Var(\theta) = TBA \tag{1}$$

$$\mathbb{E}(\theta|X = x) = g_1(\mu, \sigma^2, x, d) = TBA \tag{2}$$

$$Var(\theta|X = x) = g_2(\mu, \sigma^2, x, d) = TBA \tag{3}$$

2.1.3 Selecting next sample depth

The key advantage of the approach given in this work over the previous work, [], is the dynamic sampling scheme which utilises the current belief to minimise the work to obtain an estimate. To do this we look to maximise the information gain per sample or equivalently maximise the reduction in variance of our belief from each sample. The decision variable throughout is circuit depth for the next sample, i.e. the number applications of the Grover operators.

To do this we utilise the expected reduction in variance, see equation **??**. Given our current state , (μ, σ^2), we select the depth that minimises the equation 1, note the optimisation is over integers. Rather than directly minimise this instead we look to maximise the variance reduction factor,

$$h(\mu, \sigma^2, d) =?? \tag{4}$$

which is equivalent to minimising the expected variance. See figure **??** for an example of this function which is typical and integer points for which we are maximising over. There are a couple of features to note:

- The various reduction function is bounded ... which is maximised by .... This reflects the scaling seen in the fixed strategy and in a number of works relating to quantum phase estimation [].
- The true maximum of integers might be a distance from the maximum of the bounding function due to the sinusoidal nature. Importantly it is not necessarily either integer value which neighbours the bounding maximum.
- The issue with selecting the bounding maximum is partially observed in this work, []. The driving factor is the gradient of the probability function for the Bernoulli distribution, near probability 0 and 1 this is small but nearer 1/2 this maximised

which improves ability discriminate different values of θ. This is something that can be clearly be observed in our experiments, see Section **??** and figure **??**.

2.1.4 Adjustments for noise

When we incorporate decoherence noise, we incorporate an extra parameter λ which represents the noise level of the quantum computer. We assume in this work that the value of this parameter has been previously ascertained through other means. For reference, we repeat here the probability of measuring a value 1 as a function of θ and λ,

$$\mathbb{P}(X = 1|\theta, \lambda) = Tobeaddedtobeconsistent.$$

For more further details of this see Section **??**.

As mentioned previously the basics sampling algorithm does not change, only the functions h, g_1, g_2 change to accommodate the change in the measurement probability. These become

$$h'(\mu, \sigma^2, d) = TBA \tag{5}$$
$$g_1'(\mu, \sigma^2, x, d) = TBA \tag{6}$$
$$g_2'(\mu, \sigma^2, x, d) = TBA \tag{7}$$

For the derivation see Appendix **??**.

2.1.5 Modelling the sampling scheme

It is useful to note that we can model the algorithm as a Markov process on $\mathbb{R}^2$ with the state as (μ, σ^2). We can see this from equations 2 and 3, since the state update only depends on the current state, the depths which is again only a function of the current state and the random sample. This means that it obeys the Markov property, giving use a Markov process. If we look closely at equation 3 we note that the update is multiplicative in nature. By taking the log of the variance, $\nu = \log(\sigma^2)$, we change this to additive process which provides us with a number of insights.

Firstly, if we consider the expected decrease in ν by taking the log of equation 1 at the optimal depth we find this is $o(1)$, see Appendix **??** through out. This means that the number of circuit executions to reach an accuracy of order ϵ will be $-\log(\epsilon)$ and ν decreases linearly in the number of iterations. This is something we observe in the experimentations, see Section **??**.

Secondly, by consider ν we can see why we need to calculate analytically the change in variance rather than use a sampling scheme. It is well known that the sample variance is an unbiased estimator of the true variance, i.e. $\mathbb{E}(S^2) = \sigma^2$ where S^2 is the sample variance, which would indicate that we can use a sample from the posterior distribution to estimate the variance. Unfortunately, this is not true as the $\log(S^2)$ is not an unbiased estimator of $\log(\sigma^2)$, we have

$$\mathbb{E}(\log\Big(S^2\Big)) = \mathbb{E}(\log\Big(\sigma^2\Big)) + \kappa,$$

where $\kappa > 0$. For details, see Appendix **??**. This means that if the sample variance is used to estimate the variance of the posterior we have a systematic drift which means the uncertainty is underestimated. This is not so problematic in the noiseless case as all steps of ν are order 1 but in the case with noise the step size decays with ν so the systematic drift dominates leading to under sampling.

Note: Can we prove something about optimality of greedy strategy given the maximum reduction factor and bounds on the value function? (we know bounded between inverse square and inverse)

2.2 Post processing

As mentioned previously the use of the normal approximation for the posterior distribution can lead to a drift in the estimate of θ. For reason we carry out a post-processing step which utilises all the obtained samples by estimating θ by the maximum likelihood estimate (MLE) as described in []. There are two important points to note about this:

1. The use of the MLE in this case is not turning our back on the Bayesian approach in the rest of the paper. Here assuming the prior distribution is uniformly distributed over the interval $[0, \pi]$, an uninformative prior, the mode of the posterior distribution, a standard point estimate in Bayesian statistics [], agrees with the MLE.

2. This approach is easily adapted to the case of decoherence noise. The likelihood is changed to incorporate the extra factor in obvious fashion.

ADD ALG AS A REMINDER

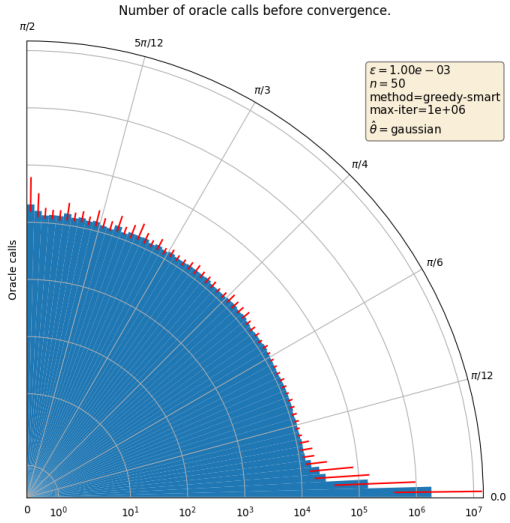
3 Experimentation

We conduct a series of experiments for the noiseless and noisy simulations of our algorithm in this section. We empirically compare against the IAE and MLAE algorithms. As the update steps for these algorithms are equivalent to sampling from a Bernoulli distribution for a given choice of θ_0 and noise level we sample directly from the Bernoulli distribution instead of simulating the dynamics of a quantum system for each algorithm. This algorithm has also been implemented and tested using Qiskit [13], with a view for future experiments with quantum simulators and quantum hardware.

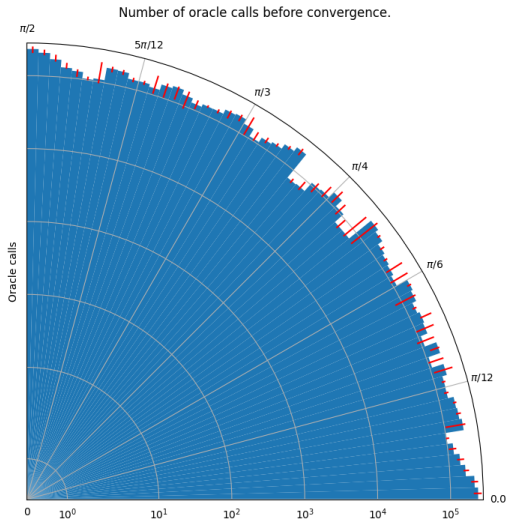
3.1 Simulation

3.1.1 Noiseless

First, we explore the convergence time for a range $\theta \in [0, \pi/2]$. As shown in [14], so-called exceptional points with poor convergence occur near to rational multiples of π, therefore we sample values of θ from continuous distributions instead of equally spaced angles. Figure 3a shows that the convergence for θ close to 0 is several orders of magnitude larger than convergence for other values. For this reason, we consider partitioning the space $\Theta = [0, \pi/2]$ into the central region $\Theta_0 = [\frac{\pi}{12}, \frac{5\pi}{12}]$ and the edge region $\Theta_1 = \Theta \backslash \Theta_0 = [0, \frac{\pi}{12})$. Note that this is not the case for some



(a) Median number of oracle calls for SAE to converge to a precision of $\varepsilon = 10^{-3}$, with 25% and 75% quartiles shown. Each bar is a region of width $\pi/120$ with the median number of oracle calls calculated from 50 values of true value θ_0 selected uniformly at random. The prior for each iteration is taken to be $N(\tilde{\theta}_0, 1)$, where $\tilde{\theta}_0$ is sampled from a $N(\theta_0, 0.1)$ distribution and success probability $1 - \alpha$ with $\alpha = 0.01$.



(b) Median number of oracle calls for IAE to converge to a precision of $\varepsilon = 10^{-3}$, with 25% and 75% quartiles shown. Each bar is a region of width $\pi/120$ with the median number of oracle calls calculated from 50 values of true value θ_0 selected uniformly at random, and a success probability $1 - \alpha$ with $\alpha = 0.01$

(The following is all conjecture as we don't have a comparison graph.)
As seen in Figure **??**, the set Θ_1 performs poorly for all methods that take a statistical approach. This region contains points where the variance reduction factor approaches 0, with the extreme points where $\theta = 0, \pi$.

From now on, we restrict to considering $\theta \in \Theta_0$.

[cdv 5] Misses a caption of the 'full' figure.

[cdv 6] It's actually from 1000 values. I can change that either in the figure, or in the caption.

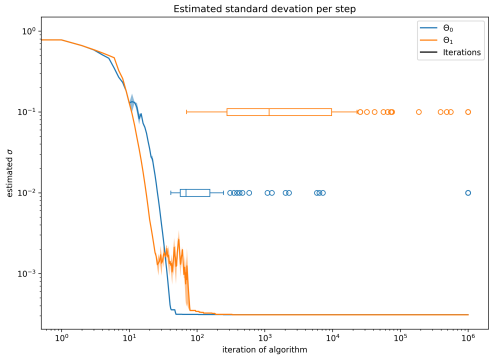


Figure 4: Median standard deviation for fixed error rate $\varepsilon = 10^{-3}$ against time step for $\theta_0 \in \Theta_0$ and $\theta_0 \in \Theta_1$. We sample θ_0 from a uniform distribution of x and 50 samples for Θ_0 and Θ_1 respectively. The prior for each iteration is taken to be $N(\theta_0, 1)$ and success probability $1 - \alpha$ with $\alpha = 0.01$. The box plots show the Q1, Q2 and Q3 algorithm termination times for the recorded runs.

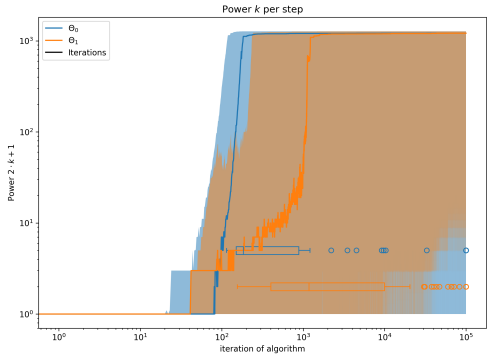


Figure 5: Median depth for fixed error rate $\varepsilon = 10^{-3}$ against time step for $\theta_0 \in \Theta_0$ and $\theta_0 \in \Theta_1$. We sample θ_0 from a uniform distribution of x and 50 samples for Θ_0 and Θ_1 respectively. The prior for each iteration is taken to be $N(\theta_0, 1)$ and success probability $1 - \alpha$ with $\alpha = 0.01$. The box plots show the Q1, Q2 and Q3 algorithm termination times for the recorded runs.

A source of potential error in this algorithm is the normal approximation made at each step, which removes the theoretical guarantees of the Bernstein-von-Mises theorem. This can be seen in Figure 6, where should expect $(1-\alpha)\%$ of the points to lie above the diagonal. As a mitigation technique, we post-process the measurement data using the MLE estimator, and recover some of the desired accuracy.

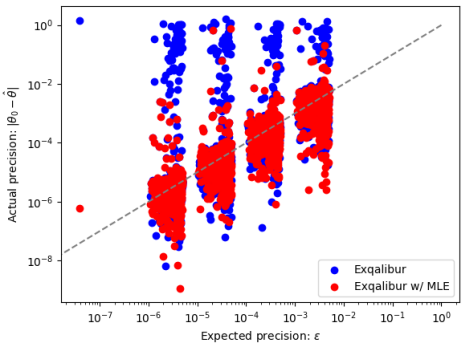


Figure 6: Actual precision of the final estimate for SAE with and without MLE post-processing. We sample $\theta_0 \in \Theta_0$ from a uniform distribution of x and 50 for target precisions of $\varepsilon = 10^{-3}, 10^{-4}, \ldots, 10^{-7}$. The prior for each iteration is taken to be $N(\theta_0, 1)$ and success probability $1-\alpha$ with $\alpha = 0.01$.

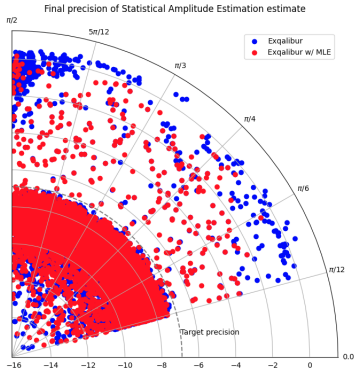


Figure 7: Actual precision of the final estimate for SAE with and without MLE post-processing. We sample $\theta_0 \in \Theta_0$ from a uniform distribution of x and 500 for a target precision of $\varepsilon = 10^{-3}$. The prior for each iteration is taken to be $N(\theta_0, 1)$ and success probability $1-\alpha$ with $\alpha = 0.01$.

We compare the performance of SAE to IAE and MLAE, and observe that in the noiseless case we achieve similar query to IAE. However, there still remain some points for which SAE fails to satisfactorily converge.

For the SAE routine, we also calculate the expected and exact probability of the Bernoulli sample at each stage. 10 is a typical example of a SAE run, where the probability is initially expected to be 0 for a substantial amount of time, before approaching $\frac{1}{2}$ for the remainder of the run.

3.1.2 Decohering Noise

We explore the performance of our algorithm for a range of noise rates.

As shown in Figure 11, the algorithm will not go beyond a depth of $\sim \frac{1}{\lambda}$.

- Concerned that error comparisons make no sense here. These algorithms have no ca-

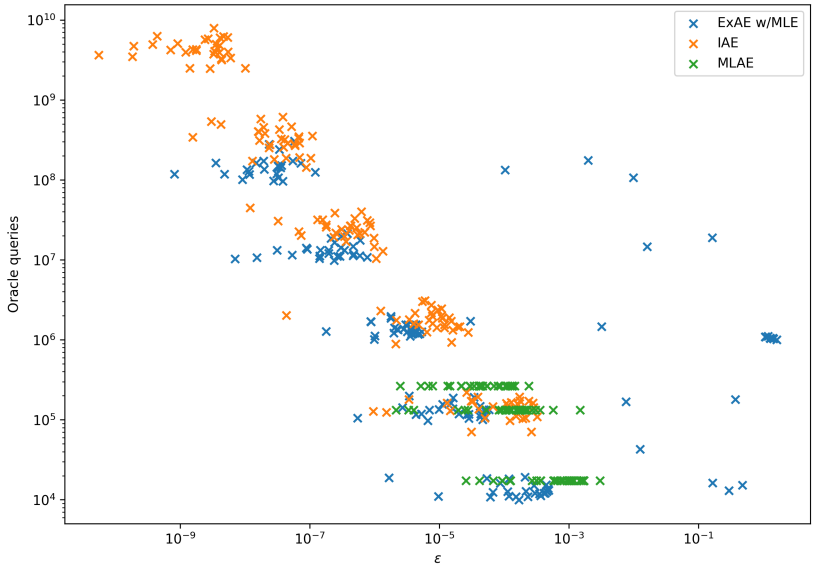


Figure 8: Actual precision versus total number of oracle queries used for IAE, MLAE and SAE. We target precisions of $\varepsilon = 10^{-3}, 10^{-4}, \ldots, 10^{-7}$, with each point a single value of $\theta_0 \in \Theta_0$ and take the final precision to be half the width of the confidence interval. We sample uniformly from Θ_0 for 30 values of θ_0. Each value of θ_0 is evaluated for all algorithms and each target ε. The prior for each iteration is taken to be $N(\theta_0, 1)$ and success probability $1 - \alpha$ with $\alpha = 0.01$.

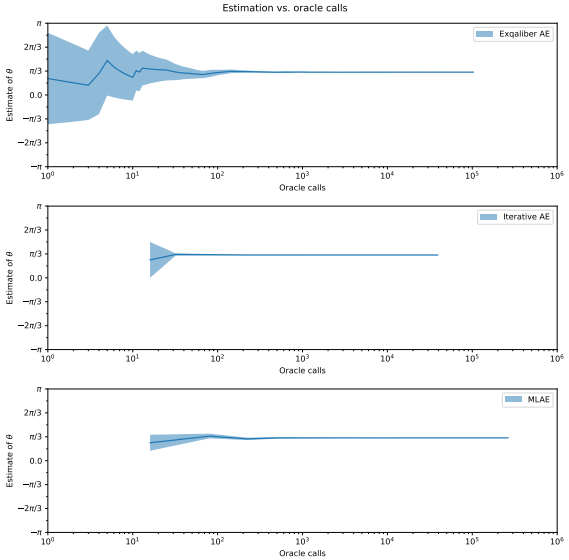


Figure 9: Confidence intervals for a single run of SAE, IAE and MLAE with $\theta_0 = 1$. The prior for SAE is taken to be $N(\pi/2, 1)$ and success probability $1 - \alpha$ with $\alpha = 0.01$. As IAE and MLAE use a fixed number of shots per step, the number of oracle calls does not start at 0.

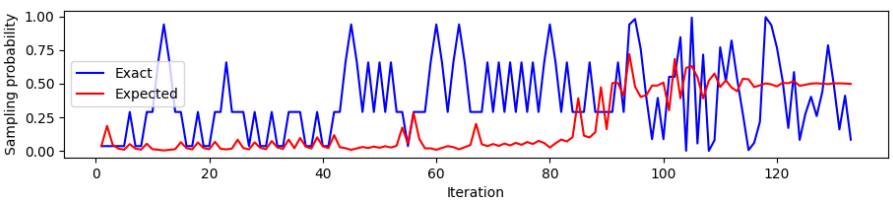


Figure 10: Exact and expected probability calculation for a single run of SAE.

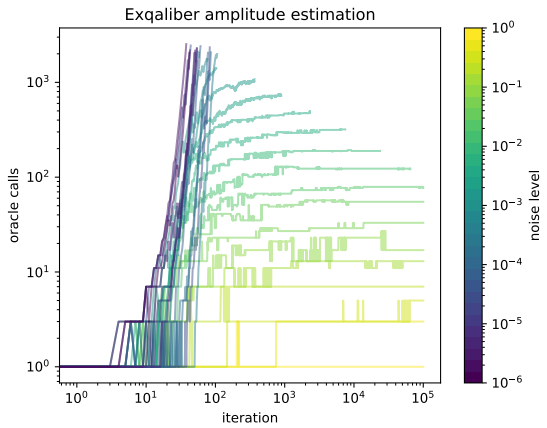


Figure 11: Depth of SAE for $\theta_0 = 1$ at varying levels of decohering noise characterised by $\lambda = 10^0, 10^{-1}, \ldots 10^{-6}$. The prior for each iteration is taken to be $N(\pi/2, 1)$, we target $\varepsilon = 10^{-3}$ and success probability $1 - \alpha$ with $\alpha = 0.01$.

Figure 12: Total number of oracle queries for SAE with decohering noise characterised by $\lambda = 10^{-1}, 10^{-2}, \ldots, 10^{-7}$. Each point is a single value of $\theta_0 \in \Theta_0$. We sample uniformly from Θ_0 for 30 values of θ_0. Each value of θ_0 is evaluated for all decohering noise values and each target ε. The prior for each iteration is taken to be $N(\theta_0, 1)$ and success probability $1 - \alpha$ with $\alpha = 0.01$.

pacity to cope with noise so obviously they're bad - but others e.g. Hitachi, QoPrime, Power-law AE do

Figure 13: Final precision versus total number of oracle queries used for IAE, MLAE and SAE with decohering noise characterised by $\lambda = 10^{-3}$. We target precisions of $\varepsilon = 10^{-3}, 10^{-4}, \ldots, 10^{-7}$, with each point a single value of $\theta_0 \in \Theta_0$. We sample uniformly from Θ_0 for 30 values of θ_0. Each value of θ_0 is evaluated for all algorithms and each target ε. The prior for each iteration is taken to be $N(\theta_0, 1)$ and success probability $1 - \alpha$ with $\alpha = 0.01$.

4 Discussion

This is for internal discussions not external release Topics to include here include:

- Directions for future including reinforcement learning
- Superconducting and trapped ion thought experiment
- Discussion about general dynamic programming approach for considering error mitigation and correction
- Connection to QPE and applicability of thinking to that setting (maybe include small section on the more complicated statistical challenge of general state)
- Circular distributions and what we know about them (probably subsection of this section or algorithm)

References

[1] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. "Quantum amplitude amplification and estimation". Contemporary Mathematics **305**, 53–74 (2002).

[2] A Yu Kitaev. "Quantum measurements and the abelian stabilizer problem" (1995). arXiv:quant-ph/9511026.

[3] Don Coppersmith. "An approximate fourier transform useful in quantum factoring" (2002). arXiv:quant-ph/0201067.

[4] Scott Aaronson and Patrick Rall. "Quantum approximate counting, simplified". Pages 24–32. Society for Industrial and Applied Mathematics. (2020). arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611976014.5.

[5] Yohichi Suzuki, Shumpei Uno, Rudy Raymond, Tomoki Tanaka, Tamiya Onodera, and Naoki Yamamoto. "Amplitude estimation without phase estimation". Quantum Information Processing **19**, 1–17 (2020).

[6] Chu-Ryang Wie. "Simpler quantum counting" (2019). arXiv:1907.08119.

[7] Paul Dagum, Richard Karp, Michael Luby, and Sheldon Ross. "An optimal algorithm for monte carlo estimation". SIAM Journal on computing **29**, 1484–1496 (2000). arXiv:https://doi.org/10.1137/S0097539797315306.

[8] Stefan Heinrich. "Quantum summation with an application to integration". Journal of Complexity **18**, 1–50 (2002).

[9] Ashley Montanaro. "Quantum speedup of monte carlo methods". Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **471**, 20150301 (2015). arXiv:https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.2015.0301.

[10] Dmitry Grinko, Julien Gacon, Christa Zoufal, and Stefan Woerner. "Iterative quantum amplitude estimation". npj Quantum Information **7**, 52 (2021). arXiv:https://doi.org/10.1038/s41534-021-00379-1.

[11] Tudor Giurgica-Tiron, Iordanis Kerenidis, Farrokh Labib, Anupam Prakash, and William Zeng. "Low depth algorithms for quantum amplitude estimation". Quantum **6**, 745 (2022). arXiv:https://doi.org/10.22331/q-2022-06-27-745.

[12] Joseph G. Smith, Crispin H. W. Barnes, and David R. M. Arvidsson-Shukur. "An adaptive bayesian quantum algorithm for phase estimation" (2023). arXiv:2303.01517.

[13] Qiskit contributors. "Qiskit: An open-source framework for quantum computing" (2023).

[14] Adam Callison and Dan E. Browne. "Improved maximum-likelihood quantum amplitude estimation" (2022). arXiv:2209.03321.

A Technical details

Details of technical work if needed