

# Efficient Bayesian Phase Estimation

Nathan Wiebe<sup>1</sup> and Chris Granade<sup>2,3</sup>

<sup>1</sup>*Quantum Architectures and Computation Group, Microsoft Research, Redmond, WA (USA)*

<sup>2</sup>*Centre for Engineered Quantum Systems, Sydney, NSW (Australia)*

<sup>3</sup>*School of Physics, University of Sydney, Sydney, NSW (Australia)*

We provide a new efficient adaptive algorithm for performing phase estimation that does not require that the user infer the bits of the eigenphase in reverse order; rather it directly infers the phase and estimates the uncertainty in the phase directly from experimental data. Our method is highly flexible, recovers from failures, and can be run in the presence of substantial decoherence and other experimental imperfections and is as fast or faster than existing algorithms.

## I. INTRODUCTION

Eigenvalue estimation has been a cornerstone of physics since the dawn of spectroscopy. In recent years, ideas from quantum information have revolutionized the ways that we estimate these values by providing methods that require exponentially fewer experiments than statistical sampling and total experimental time that saturates the Heisenberg limit, which is the best possible scaling allowed by quantum mechanics. This quantum approach, known as phase estimation (PE), is crucial to achieve many of the celebrated speedups promised by quantum computing [1–5].

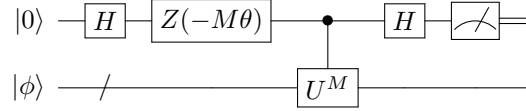
The need for efficient PE algorithms is especially poignant for Shor’s algorithm [1], wherein the unknown eigenphases must be estimated to hundreds of bits of precision. In order for the algorithm to be practical, the PE algorithm must only use a logarithmic number of measurements and the classical processing used to infer the phase must also be efficient. This excludes many existing PE methods that achieve logarithmic scaling. Other methods are both classically and quantumly efficient [6–9], but often fail to extract information about the phase as efficiently as their classically inefficient brethren.

Here we address this gap by proposing a new approach to iterative phase estimation that is based on a new approximate inference method that we call *rejection filtering* (RF). Similarly we call the PE algorithm that follows from RF, *RFPE*. RFPE is classically efficient, requires a logarithmic number of operations, saturates the Heisenberg bound, is robust to experimental imperfections such as depolarizing noise and small systematic errors, reports the uncertainty in the estimate of the phase, can track time-dependent phases and can be employed in a memory limited environment like in an FPGA.

Before going into detail about our algorithm we will first review the phase estimation circuit and Bayesian approaches to PE. We then show that rejection sampling can be used to efficiently approximate Bayesian inference, thereby allowing PE to inherit the speed and robustness of Bayesian approaches while retaining the efficiency of traditional methods.

Iterative PE infers the eigenvalue of a given eigenvector of a unitary matrix  $U$  from a set of experiments that are

performed on the circuit



where  $U|\phi\rangle = e^{i\phi}|\phi\rangle$  for an unknown eigenphase  $\phi \in \mathbb{R}$ , and where  $Z(M\theta) = e^{iM\theta Z}$  is a rotation that can be used to compare the eigenphase of  $U$  to a known reference value. Most PE algorithms use this circuit to infer the bits in a binary expansion of  $\phi$ . The process can be made near optimal (up to  $\log^*$  factors [9]) and affords an efficient inference method for these bits.

## II. BAYESIAN PHASE ESTIMATION

Bayesian phase estimation involves performing a set of experiments and then updating the prior distribution using Bayes’ rule. For example, if an experiment is performed with using  $M$  repetitions of  $U$ ,  $Z(M\theta)$  and a measurement outcome  $E \in \{0, 1\}$  is observed then Bayes’ rule states that the *posterior probability* distribution for  $\phi$  after observing the datum is

$$P(\phi|E; \theta, M) = \frac{P(E|\phi; \theta, M)P(\phi)}{\int P(E|\phi; \theta, M)P(\phi)d\phi}. \quad (1)$$

The likelihood function for phase estimation,  $P(0|\phi; \theta, M)$ , is

$$P(0|\phi; \theta, M) = \frac{1 + \cos(M[\phi + \theta])}{2}, \quad (2)$$

$$P(1|\phi; \theta, M) = \frac{1 - \cos(M[\phi + \theta])}{2}.$$

After using (1) to update the posterior distribution we then set the prior distribution to equal the posterior distribution. This process is then repeated for each of the experiments in the data set.

Unlike conventional methods, Bayesian inference returns a posterior distribution over the phase. The mean and standard deviation of this distribution provide an estimate of the true eigenvalue and the algorithm’s uncertainty in that value. More sophisticated estimates of uncertainty, such as credible regions, can also be extracted

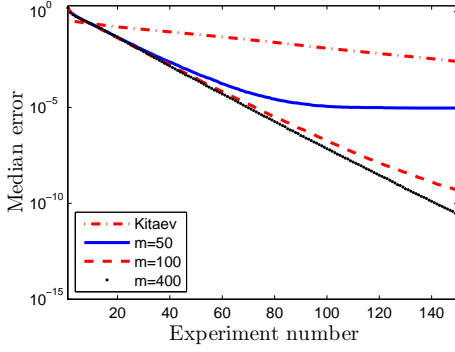


FIG. 1. Median errors in phase estimation for 10 000 random initial choices of the true eigenphase.

from the posterior distribution [10, 11]. Estimates of the uncertainty are crucial here because we use them to design highly informative experiments and also because they allow the protocol to be terminated when a threshold accuracy has been reached.

Figure 3 examines the performance of RFPE as a function of the total evolution number of applications of  $U$  needed to achieve an error threshold. The data conforms to the  $O(1/T)$  scaling expected from the Heisenberg bound. Further details are given in the appendix.

### III. APPROXIMATE BAYESIAN PHASE ESTIMATION

Exact Bayesian inference is impossible if the eigenphases are continuous so approximations are needed to make the procedure tractable. Rather than naïvely discretizing the prior distribution, modern methods discretize by sampling from the prior and then perform Bayesian inference on the discrete set of samples (often called “particles”). These particle filter methods methods are quite powerful and have become a mainstay in computer vision and machine learning [12–14].

Despite the power of these methods, they are often hard to implement and even harder to deploy in a memory limited environment such as on an embedded controller or an FPGA. With the increasing use of FPGAs in the control of quantum information experiments [15–17], overcoming this limitation presents a significant advantage to modern experiment design.

We propose a much simpler approach that we call Rejection Filtering Phase Estimation (RFPE). Rather than using a set of hypotheses that implicitly define a model for the system, we posit a prior model and directly update it to find a model for our posterior distribution. We achieve this by using a Gaussian with mean  $\mu$  and variance  $\sigma^2$  to model our initial prior, perform a Bayesian update on samples drawn from the distribution and then refit the updated samples to a Gaussian. This strategy is used in a number of particle filter methods such as

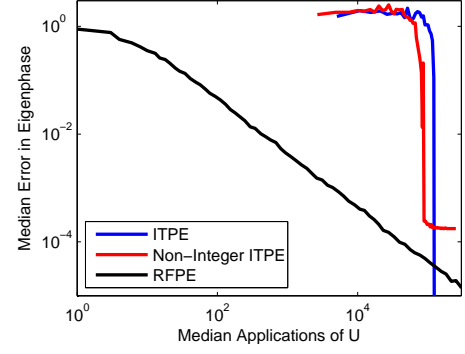


FIG. 2. Comparison of RFPE to ITPE for  $t = 10\,000$  with 100 samples for  $\phi_{\text{true}} = 2\pi k/t$  at each measurement.

the extended Kalman filter and assumed density filtering [12, 18]. We further optimize this process by approximating the Bayesian inference step using rejection sampling. This can reduce the memory required by a factor of 1 000 or more, as we need only consider a single sample at a time. Our algorithm is described below and pseudocode is given in Appendix E.

1. Perform experiment for given  $\theta$ ,  $M$  and observe outcome  $E \in \{0, 1\}$ .
2. Draw  $m$  samples from  $\mathcal{N}(\phi|\mu, \sigma^2)$ .
3. For each sample,  $\phi_j$ , assign  $\phi_j$  to  $\Phi_{\text{accept}}$  with probability  $P(E|\phi_j; \theta, M)/\kappa_E$ , where  $\kappa_E \in (0, 1]$  is a constant s.t.  $P(E|\phi_j; \theta, M)/\kappa_E \leq 1$  for all  $\phi_j, E$ .
4. Return  $\mu = \mathbb{E}(\Phi_{\text{accept}})$  and  $\sigma = \sqrt{\mathbb{V}(\Phi_{\text{accept}})}$ .

The resultant samples are equivalent to those drawn from the posterior distribution  $P(\phi|E; M, \theta)$ . To see this, note that the probability density of a sample being accepted at  $\phi = \phi_j$  is  $P(E|\phi; \theta, M)\mathcal{N}(\phi|\mu, \sigma^2)$ . Eqn (1) then implies

$$P(E|\phi; \theta, M)\mathcal{N}(\phi|\mu, \sigma^2) \propto P(\phi|E; \theta, M), \quad (3)$$

which implies that the accepted samples are drawn from the posterior distribution.

Although it is difficult to concretely predict the value of  $m$  needed to make the error in the inference small, we show in Appendix C that  $m$  must scale at least as the inverse square of the relative fluctuations in the likelihood function. Similarly, Markov’s inequality shows that  $m$  must scale at least as  $\kappa_E / \int P(E|\phi; \theta, M)P(\phi)d\phi$  to ensure that, with high probability, the mean can be accurately estimated from the accepted samples. We introduce  $\kappa_E$  to compensate for small expected likelihoods.

The main issue that remains is how to optimally choose the parameters  $\theta$  and  $M$ . One approach is to locally optimize the Bayes risk [10], but the resulting calculation can be too expensive to carry out in online experiments that provide experimental results at a rate of tens of megahertz or faster. Fortunately, the particle guess heuristic

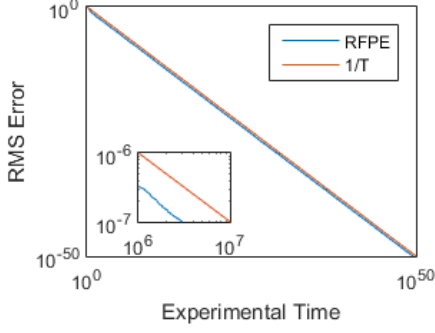


FIG. 3. Error vs evolution time for rejection filter phase estimation for the case where  $M$  is continuous and can be interpreted as the experimental time taken.

(PGH) can give an expedient and near-optimal experiment for this class of likelihood functions [19],

$$M = \left\lceil \frac{1.25}{\sigma} \right\rceil, \quad \theta \sim P(\phi). \quad (4)$$

The constant 1.25 is empirically chosen to minimize the cost of RFPE.

Figure 1 shows the error incurred using RFPE. The most obvious feature is that the error shrinks exponentially with the number of experiments (which is proportional to the evolution time under the PGH) for  $m > 100$ . Roughly 150 experiments are needed for the method to provide 32 bits of accuracy in the median case. We discuss the scaling in the mean in Appendix A.

The number of experiments needed to reach error  $\epsilon$  scales as  $O(\log(1/\epsilon))$  rather than  $O(\log(1/\epsilon) \log \log(1/\epsilon))$  in Kitaev's method [6, 7]. Concretely, after 150 experiments the median error for Kitaev's PE algorithm (with  $s = 10$ ) is roughly 10 million times that of RFPE. This  $O(\log(1/\epsilon))$  scaling is further observed over 100 orders of magnitude in  $\epsilon$  in the appendix.

Figure 2 compares RFPE to the information-theory PE (ITPE) method of Svore *et al.*. Although ITPE is inefficient and is non-adaptive, it is exact and so it is a natural benchmark to compare RFPE against. We find ITPE requires nearly five times the applications of  $U$  if  $\phi = 2\pi k/t$  for integer  $k < t$  and  $t = 10\,000$ .

Conversely, ITPE requires only 25 measurements to identify the phase with 50% probability whereas RFPE requires 51 experiments if  $k$  is an integer. If the true value of  $k$  is real-valued and ITPE is left unmodified, then ITPE fails to learn in the median because the long evolution times chosen lead to contradictory possibilities that prevents ITPE from converging to a unique answer. We correct this by choosing  $M \rightarrow \lceil M/2 \rceil$  in such cases, which increases the number of experiments to 35 but also reduces the experimental time below that of unmodified ITPE. This shows that all three methods tradeoff experimental and computational resources differently. This is especially salient because ITPE is inefficient unlike RFPE.

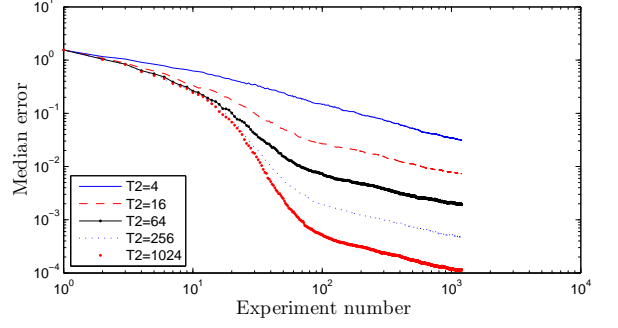


FIG. 4. Median errors for phase estimation in decohering systems for experiments constrained to use  $M \leq T_2$ . We take  $m = 12\,000$  and use 1 000 random samples to generate the data above. The initial state is taken to be a randomly chosen eigenstate in all cases.

#### A. Phase estimation with depolarizing noise

A criticism that has been levied lately at PE methods is that they can be impractical to execute on non-fault tolerant quantum hardware [20–22]. This is because phase estimation attains its quadratic advantage over statistical sampling by using exponentially long evolutions. Decoherence causes the resultant phases to become randomized as time progresses, ultimately resulting in  $\lim_{M \rightarrow \infty} P(0|\phi; \theta, M) = \frac{1}{2}$ . Measurements convey no information in this limit. It similarly may be tempting to think that decoherence fundamentally places a lower limit on the accuracy of PE. Bayesian inference to can, however, estimate  $\phi$  using experiments with  $M \approx T_2$  [10].

We model the effect of decoherence on the system by assuming the existence of a *decoherence time*  $T_2$  such that

$$\begin{aligned} P(0|\phi) &= e^{-M/T_2} \left( \frac{1 + \cos(M[\phi - \theta])}{2} \right) + \frac{1 - e^{-M/T_2}}{2}, \\ P(1|\phi) &= e^{-M/T_2} \left( \frac{1 - \cos(M[\phi - \theta])}{2} \right) + \frac{1 - e^{-M/T_2}}{2}. \end{aligned} \quad (5)$$

This model is appropriate when the time required to implement the controlled operation  $\Lambda(U)$  is long relative the that required to perform  $H$  and an arbitrary  $Z$ -rotation, as is appropriate in quantum simulation. For simplicity, in the following we assume that  $T_2$  is known.

Since  $T_2$  places a limitation on our ability to learn, we propose a variation to (4),

$$M = \min \left\{ \left\lceil \frac{1.25}{\sigma} \right\rceil, T_2 \right\}. \quad (6)$$

The experiments yielded by (6) are qualitatively similar to locally optimized experiments for frequency estimation which are chosen to saturate the Cramér-Rao bound [23]; however, (6) requires nearly 100 fold less computing time to select an experiment. We choose  $M$  to cut off at  $T_2$  in part because the locally optimized solutions

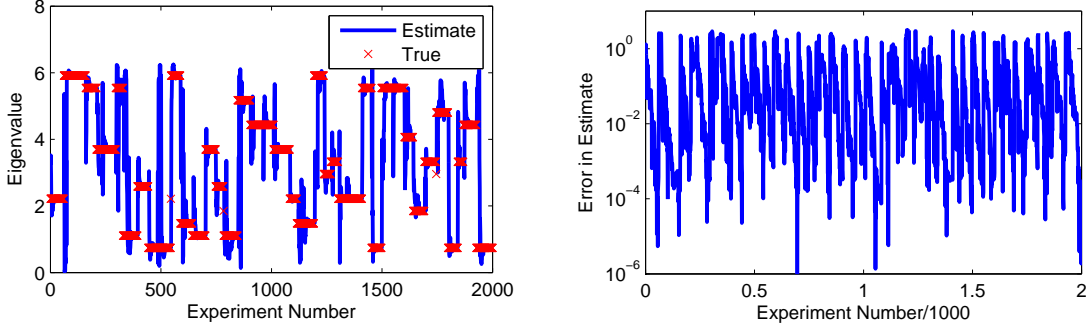


FIG. 5. Instantaneous estimate of the eigenphase for a system with 16 eigenvalues,  $\Delta = 0$ ,  $\tau = 0.1$  and  $T_2 = 10^4$ .

choose to saturate at this value which can be understood using the following argument. The Cramér-Rao bound for frequency estimation scales as  $O(M^{-2})$  [24] in the absence of decoherence. Eqn. (5) suggests that decoherence causes the posterior variance to increase as  $O(\exp(2M/T_2))$ . As in the frequency estimation case [23], calculus reveals that  $M = T_2$  optimally trades off these tendencies, which justifies (6).

Figure 4 shows that RFPE smoothly transitions between the exponential scaling expected at short times and the polynomial scaling expected when decoherence becomes significant. The error scales roughly as  $1/N^{0.6}$  in this polynomial regime, which is comparable to the  $1/\sqrt{N}$  scaling expected from statistical sampling. Decoherence therefore does not necessarily impose a fundamental limitation on the accuracy that PE can achieve. Figure 4 also shows that confining the experiments to stay in a more coherent regime also can actually hurt the algorithm's ability to infer the phase, as expected.

If we instead cost the algorithm by the total number of applications of  $U$  needed to learn the phase then we recover the (or in the case where  $M$  is continuous, the total evolution time required). In this case, we expect from the Heisenberg bound that the root-mean-square error should optimally scale as  $1/M$ . We see in figure that, over 50 orders of magnitude in the desired error tolerance, this scaling is saturated for RFPE.

#### IV. TRACKING EIGENPHASES

Figure 4 shows the performance of RFPE when the initial quantum state is an eigenstate and is discarded after each experiment. Performing phase estimation in this way minimizes the number of experiments, but can be prohibitively expensive if preparation of the initial eigenstate is prohibitively costly. In such cases, it makes sense to follow the standard prescription for phase estimation by keeping the quantum state until it is clear that the initial eigenstate has been depolarized. These depolarizations can cause RFPE to become confused because the new data that comes in is only consistent with hypotheses that have been ruled out. We address this

by performing inexpensive experiments to assess whether the state has depolarized and then restart the learning process. Restarting can even be valuable when  $T_2 = 0$  if the input state is a superposition of eigenvectors or to recover if the RFPE becomes stuck (see Appendix A).

The following procedure addresses this issue in cases where the spectral gaps are promised to be at least  $\Delta$ .

1. After each update with probability  $e^{-M/T_2}$  perform an experiment with  $\theta = \mu$  and  $M = \tau/\sigma$  for  $\tau < 1$ .
2. If result = 1 then prepare initial state and reset  $\sigma$ .
3. After restart, continue as normal until  $\sigma < \Delta$  then set  $\sigma$  and  $\mu$  to those of the closest eigenvalue.

Steps 1 and 2 perform a one-sided test of whether the prior distribution is consistent with the current state. If the prior probability distribution is correct then the probability of measuring 0 is

$$\frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} \cos^2\left(\frac{(\mu-x)\tau}{2\sigma}\right) e^{-\frac{(\mu-x)^2}{2\sigma^2}} dx = \frac{1 + e^{-\tau^2/2}}{2}. \quad (7)$$

If  $\tau = 0.1$ , the probability of measuring 0 is approximately 0.998 and hence measuring 1 implies that the hypothesis that the prior is correct can be rejected at  $p \leq 0.002$ . A Bayesian analysis of our reset rule is given in Appendix D.

Step 3 restarts the learning process if the learning process fails. It is important, however, to not throw away the spectral information that has been learned prior to the restart. Step 3 reflects this by checking to see if the current estimate of the eigenphase corresponds to a known eigenstate and then sets  $\mu$  to be the estimate of the eigenvalue and  $\sigma$  to be its uncertainty if such an identification is made. This allows the algorithm to resume learning once the depolarized state is projected onto a known eigenstate.

The test process also permits the eigenvalue of an eigenstate in a decohering system to be estimated in real time. Figure 5 shows that the restarting algorithm can rapidly detect a transition away from the instantaneous eigenstate and then begin inferring the eigenvalue of the system's new instantaneous eigenstate.

## V. CONCLUSION

Our work makes phase estimation more experimentally relevant by substantially reducing the experimental time required and also by making the process resilient to decoherence. This is especially critical as current experiments push past the classical regime and IPE becomes increasingly impractical in lieu of fault-tolerance. In particular, the ability of our algorithm to learn in the presence of decoherence provides an efficient alternative to the variational eigensolvers used in present day experiments [20–22].

Looking at the problem of phase estimation more generally, it is clear that it is in essence an inference problem. Our work shows that modern ideas from machine learning can be used therein to great effect. It is our firm belief that by combining the tools of data science with clever experimental design, further improvements can be seen not only in phase estimation but also quantum metrology in general.

## ACKNOWLEDGMENTS

We thank B. Terhal, K. Rudinger and D. Wecker for useful comments.

- 
- [1] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM Journal on Computing* **26**, 1484 (1997).
  - [2] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, “Quantum amplitude amplification and estimation,” *Contemporary Mathematics* **305**, 53 (2002), quant-ph/0005055.
  - [3] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, “Simulated quantum computation of molecular energies,” *Science* **309**, 1704 (2005).
  - [4] A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations,” *Physical Review Letters* **103**, 150502 (2009).
  - [5] B. P. Lanyon, J. D. Whitfield, G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, *et al.*, “Towards quantum chemistry on a quantum computer,” *Nature Chemistry* **2**, 106 (2010).
  - [6] A. Y. Kitaev, “Quantum measurements and the Abelian stabilizer problem,” *Electronic Colloquium on Computational Complexity* **3** (1996).
  - [7] A. Y. Kitaev, A. Shen, and M. N. Vyalov, *Classical and quantum computation*, Vol. 47 (American Mathematical Society Providence, 2002).
  - [8] B. L. Higgins, D. W. Berry, S. D. Bartlett, H. M. Wiseman, and G. J. Pryde, “Entanglement-free Heisenberg-limited phase estimation,” *Nature* **450**, 393 (2007).
  - [9] K. M. Svore, M. B. Hastings, and M. Freedman, “Faster phase estimation,” *Quantum Information & Computation* **14**, 306 (2014).
  - [10] C. E. Granade, C. Ferrie, N. Wiebe, and D. G. Cory, “Robust online Hamiltonian learning,” *New Journal of Physics* **14**, 103013 (2012).
  - [11] C. Ferrie, “High posterior density ellipsoids of quantum states,” *New Journal of Physics* **16**, 023006 (2014).
  - [12] S. Haykin, *Kalman filtering and neural networks*, Vol. 47 (John Wiley & Sons, 2004).
  - [13] A. Smith, A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice* (Springer Science & Business Media, 2013).
  - [14] M. Isard and A. Blake, “CONDENSATION — Conditional Density Propagation for Visual Tracking,” *International Journal of Computer Vision* **29**, 5 (1998).
  - [15] M. D. Shulman, S. P. Harvey, J. M. Nichol, S. D. Bartlett, A. C. Doherty, V. Umansky, and A. Yacoby, “Suppressing qubit dephasing using real-time Hamiltonian estimation,” *Nature Communications* **5**, 5156 (2014).
  - [16] S. Casagrande, “On design and testing of a spectrometer based on an FPGA development board for use with optimal control theory and high-Q resonators,” (2014).
  - [17] J. Hornibrook, J. Colless, I. Conway Lamb, S. Pauka, H. Lu, A. Gossard, J. Watson, G. Gardner, S. Fallahi, M. Manfra, and D. Reilly, “Cryogenic Control Architecture for Large-Scale Quantum Computing,” *Physical Review Applied* **3**, 024010 (2015).
  - [18] M. Oppen and O. Winther, “A bayesian approach to online learning,” *On-line Learning in Neural Networks*, 363 (1998).
  - [19] N. Wiebe, C. Granade, C. Ferrie, and D. Cory, “Hamiltonian learning and certification using quantum resources,” *Physical Review Letters* **112**, 190501 (2014).
  - [20] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, “A variational eigenvalue solver on a photonic quantum processor,” *Nature Communications* **5**, 4213 (2014).
  - [21] J. R. McClean, R. Babbush, P. J. Love, and A. Aspuru-Guzik, “Exploiting locality in quantum computation for quantum chemistry,” *Journal of Physical Chemistry Letters* **5**, 4368 (2014).
  - [22] D. Wecker, M. Hastings, and M. Troyer, “Towards practical quantum variational algorithms,” arXiv:1507.08969v1 (2015).
  - [23] C. Ferrie, C. E. Granade, and D. G. Cory, “How to best sample a periodic probability distribution, or on the accuracy of Hamiltonian finding strategies,” *Quantum Information Processing* **12**, 611 (2013).
  - [24] N. Wiebe, C. Granade, and D. G. Cory, “Quantum bootstrapping via compressed quantum hamiltonian learning,” *New Journal of Physics* **17**, 022005 (2015).
  - [25] N. Wiebe, C. Granade, C. Ferrie, and D. Cory, “Quantum Hamiltonian learning using imperfect quantum resources,” *Physical Review A* **89**, 042314 (2014).
  - [26] S. Goodman, “A dirty dozen: Twelve p-value misconceptions,” *Seminars in Hematology Interpretation of Quantitative Research*, **45**, 135 (2008).
  - [27] R. Hoekstra, R. D. Morey, J. N. Rouder, and E.-J. Wagenmakers, “Robust misinterpretation of confidence intervals,” *Psychonomic Bulletin & Review*, 1 (2014).



## Appendix A: Variance Reduction Strategies

An important drawback of RFPE is the tails for the distribution of errors in phase estimation can be quite fat in typical applications, meaning that there is significant probability that the error in the inferred eigenphase is orders of magnitude greater than the median. Such a distribution can be seen in [Figure 6](#) where we see that although the median error is roughly  $10^{-10}$  radians after 100 updates for  $m > 50$  a non-negligible fraction of the experiments have error on the order of 1.

Fortunately, the need to always repeat the algorithm and use a majority voting scheme to reduce the variance of the estimate is mitigated by the fact that the algorithm outputs  $\sigma$  which estimates the uncertainty in the resultant eigenphase. Alternatively, less aggressive experiments can be chosen in the guess heuristic or multi-modal models for the prior distribution (such as Gaussian mixture models) can be used. The later approach can be quite valuable in computer vision and machine learning, however here we have an additional freedom not typically enjoyed in those fields: we can perform adaptive experiments to test to see if our current hypothesis is correct.

The central idea behind our restarting strategy is to examine the decay of  $\sigma$  with the number of experiments. In ideal cases, the error decays exponentially which means that it is easy to see when the algorithm fails by plotting  $\sigma$  on a semi-log plot. Such intuition can be easily automated. The idea behind our restarting algorithm is to first estimate the derivative of  $\log(\sigma)$  with respect to the experiment number and determine whether it is less than a threshold. If it is less than the threshold, perform an experiment to test to see if the value of  $\mu$  that has been learned so far is accurate (as per our incoherent phase estimation algorithm). If it is found to be inaccurate then restart the algorithm and abandon the information learned so far. The restarting portion of this algorithm is given in [Algorithm 3](#).

When a restarting strategy like this is employed, we need to modify the model selection criteria. Previously, we used the value of  $\mu$  yielded by the most recent experiment as our estimate. Given that a restart late in the algorithm can reset all information learned about a model, it makes sense in this context to use the value of  $\mu$  corresponding to the smallest  $\sigma$  observed. This corresponds to the model that the inference algorithm has the greatest certainty.

We see in [Figure 7](#) that this strategy of restarting substantially reduces the weights of the tails. In fact, the mean error in the inference falls from 0.0513 radians to  $1.08 \times 10^{-6}$  radians. This shows that this resetting strategy substantially reduce the probability of a large error occurring in the estimate of the eigenphase. This shows that heuristic error correction methods can be effective at reducing the mean error. Previous approaches that succeeded in making the mean-error small used costly numerically optimized experiments, which render such approaches impractical for online phase estimation [[10](#), [19](#), [23–25](#)].

Note that Kitaev’s phase estimation algorithm also can have a high probability of failure throughout the algorithm, so one might wonder if these restarting strategies might also be of value for traditional phase estimation algorithms. Unfortunately, these strategies cannot be easily translated to Kitaev’s PE algorithm because the most expensive experiments are performed first. Thus, restarting does not allow the algorithm to combat decoherence, nor can it be used to inexpensively fix situations where one of the bits is inferred improperly.

Not all errors occur because of decoherence. Because rejection filtering is an approximate Bayesian method, one pathology that can occur is that the algorithm fails to learn because the approximate nature of the resampling step may exclude a region of the model space that in fact contains the true model. This is especially common when the inference algorithm is provided with a datum for which the model that has maximum apriori probability assigns low probability. In such circumstances rejection filtering can become convinced of a hypotheses that is not true and the algorithm will not continue to learn.

If the algorithm stops learning then it is seldom advisable to restart the entire algorithm from scratch. Instead, a better approach is to follow the same strategy considered above by proposing an experiment to test to see if the current uncertainty is less than the true error. If the test fails then rather than restarting the variance of the model is increased by a constant factor and the inference is continued. This protocol is taken in [Figure 3](#) wherein the variance is increased by a factor of 100 if the test fails.

## Appendix B: Stability Against Errors in the Likelihood Function

Our algorithm is not only robust against well known depolarizing noise but is robust against *uncharacterized noise sources* as well. We demonstrate this in [Figure 8](#) wherein we introduce depolarizing noise of strength  $\gamma$  to our experimental system, but do not include such noise in the likelihood function. For example, with  $\gamma = 0.4$ , the measurement outcome of phase estimation is replaced with a random bit with probability 40%. Although this may seem qualitatively similar to the numerical examples for depolarizing noise considered in the main body of the paper, this case is much more pathological because it is only used to generate the experimental data and is not permitted in the model of the system used in the likelihood function. This raises the possibility that the algorithm could become

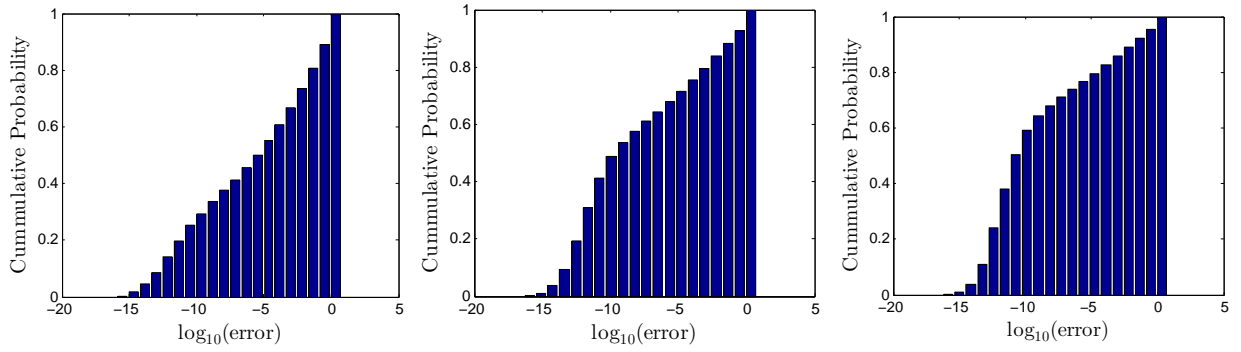


FIG. 6. Cumulative distribution function of probability that PE error is less than  $x$  after 150 updates for  $m = 50$  (left)  $m = 100$  (middle)  $m = 200$  (right).

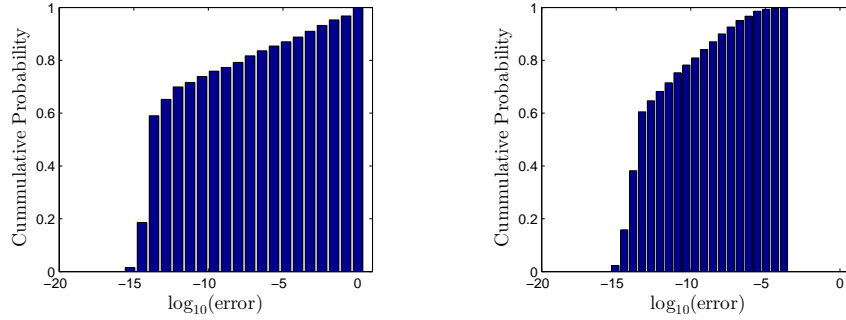


FIG. 7. Cumulative distribution function of probability that PE error is less than  $x$  after 200 updates for  $m = 2000$  for  $\Gamma = \infty$  (left)  $\Gamma = 0.1$  (right) and  $\tau = 0.1$ . Estimate of CDF consists of 1000 random eigenphase inference problems with  $T_2 = \infty$ .

confused due to experimental results that are fundamentally inconsistent with the assumed likelihood function for the system.

Figure 8 shows that the inclusion of uncharacterized depolarizing noise does not actually prevent the eigenvalue from being estimated. Rather it reduces the number of bits per experiment that the algorithm can infer. We see this by fitting the exponential part of the data in Figure 8 (along with similar data for  $\gamma = 0.1$ ,  $\gamma = 0.3$ ,  $\gamma = 0.5$ ) and find that the error decay exponent,  $\lambda$ , to shrink as roughly  $\lambda \approx 0.17e^{-3.1\gamma}$  it does not prevent our algorithm from learning at an exponential rate (until depolarizing noise becomes significant). Thus RFPE continue to work even in the presence of depolarizing noise sources that are both strong and uncharacterized.

### Appendix C: Stability of Rejection Filtering PE

One way in which the rejection sampling algorithm can break down is if the likelihood function becomes too flat relative to the number of discrete samples,  $m$ , drawn from the prior distribution. This breakdown occurs because the sample variance in the posterior mean is much greater than the difference between the prior mean and posterior means, which are not expected to be approximately the same if the likelihood function has little variation. This in effect implies that the dynamics of the mean will essentially be a random walk and as a result we do not expect that our rejection sampling method to perform well if the likelihood function is too flat.

In practice, the flatness of the distribution can be reduced by batching several experiments together and processing them. This process is in general inefficient, but can be made efficient if an appropriate value of  $\kappa_E$  is known for each datum recorded.

The question remaining is: how flat is too flat? We show in the theorem below that if the number of samples taken from the true posterior distribution does not scale at least inverse quadratically with the scale of relative fluctuations in the likelihood function then we expect the posterior variance to be much greater than the shift in the means.

**Theorem 1.** Assume that for all  $j$   $P(E|x_j) = \alpha + \delta_j$  where  $|\delta_j| \leq \delta$  and  $\alpha \geq 10\delta$  and assume that  $x_j \sim P(x_j)$ . If we then define  $\mu_0 := \sum_j P(x_j)x_j$ ,  $\mu_1 := \sum_j P(x_j|E)x_j$  and  $\sigma_1^2$  to be the posterior variance then  $|\mu_1 - \mu_0| \in \Omega(\sigma_1/\sqrt{m})$

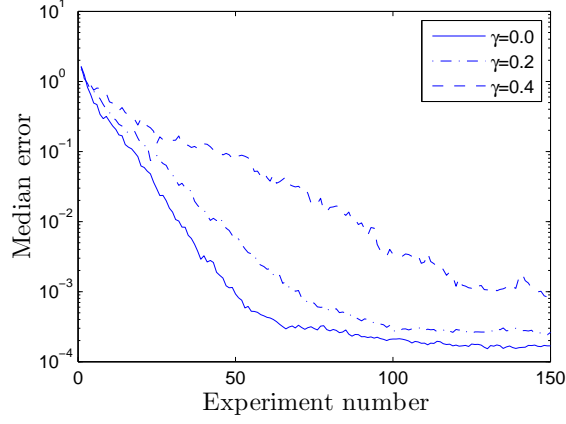


FIG. 8. Errors in inference for phase estimation with different levels of un-modeled noise,  $\gamma$ , for  $T_2 = 1000$ . In each instance the initial state was taken to be a fixed, but unknown, eigenstate of the Hamiltonian. The median was found for 100 randomly chosen values of the eigenphase of this fixed initial eigenstate.

only if  $\frac{\alpha^2}{\delta^2} \in O(m)$ .

*Proof.* Bayes' rule gives us

$$\left| \sum_j P(x_j|E)x_j - \mu_0 \right| = \left| \frac{\sum_j P(E|x_j)P(x_j)(x_j - \mu_0)}{\sum_j P(E|x_j)P(x_j)} \right| = \left| \frac{\sum_j \delta_j P(x_j)(x_j - \mu_0)}{\sum_j P(E|x_j)P(x_j)} \right|. \quad (\text{C1})$$

Then using the Cauchy–Schwarz innequality, the triangle inequality and  $\alpha \geq 2\delta$ .

$$\left| \frac{\sum_j \delta_j P(x_j)(x_j - \mu_0)}{\sum_j P(E|x_j)P(x_j)} \right| \leq \left| \frac{\delta \sqrt{\sum_j P(x_j)|x_j - \mu_0|^2}}{\alpha - \delta} \right| \leq \frac{\delta\sigma}{\alpha - \delta} \leq \frac{2\delta\sigma}{\alpha}. \quad (\text{C2})$$

Thus the maximum shift in the posterior mean shrinks as the the likelihood function becomes increasingly flat, as expected.

Next we need to lower bound the posterior variance in order ensure that the value of  $m$  chosen suffices to make the error small in the best possible case. To do so we use the reverse triangle inequality:

$$\sigma_1^2 = |\sigma_1^2 - \sigma^2 + \sigma^2| \geq \sigma^2 - |\sigma_1^2 - \sigma^2|. \quad (\text{C3})$$

Thus it suffices to upper bound  $|\sigma_1^2 - \sigma^2|$  to lower bound  $\sigma_1^2$ . To do so, note that  $\alpha \geq 2\delta$  and hence

$$|P(x_j|E) - P(x_j)| = \left| \frac{P(x_j)(\delta_j - \sum_j P(x_j)\delta_j)}{\alpha + \sum_j P(x_j)\delta_j} \right| \leq \frac{P(x_j)2\delta}{\alpha - \delta} \leq \frac{P(x_j)4\delta}{\alpha}. \quad (\text{C4})$$



Now the difference between the two variances can be written as

$$\begin{aligned}
|\sigma_1^2 - \sigma^2| &= \left| \sum_j P(x_j|E)(x_j - \mu_1)^2 - P(x_j)(x_j - \mu_0)^2 \right| \\
&\leq \left| \sum_j (P(x_j|E) - P(x_j))(x_j - \mu_1)^2 \right| + \left| \sum_j P(x_j)((x_j - \mu_0)^2 - (x_j - \mu_1)^2) \right| \\
&\leq \frac{4\delta}{\alpha} \left| \sum_j P(x_j)(x_j - \mu_1)^2 \right| + (\mu_1 - \mu_0)^2 \\
&\leq \frac{4\delta\sigma^2}{\alpha} + \frac{4\delta}{\alpha} \left| \sum_j P(x_j)[(x_j - \mu_1)^2 - (x_j - \mu_0)^2] \right| + (\mu_1 - \mu_0)^2 \\
&\leq \frac{4\delta\sigma^2}{\alpha} + (1 + \frac{4\delta}{\alpha})(\mu_1 - \mu_0)^2 \leq \frac{4\delta\sigma^2}{\alpha} + \frac{12\delta^2\sigma^2}{\alpha^2} \leq \frac{10\delta\sigma^2}{\alpha}.
\end{aligned} \tag{C5}$$

Thus we have that

$$\sigma_1^2 \geq \sigma^2(1 - 10\delta/\alpha). \tag{C6}$$

Now assuming  $\delta \leq \alpha/10$  we have

$$\sigma_1^2 \in \Omega(\sigma^2). \tag{C7}$$

Finally, we note that

$$|\mu_1 - \mu_0| \in \Omega(\sigma_1/\sqrt{m}) \Rightarrow \frac{\delta\sigma}{\alpha} \in \Omega(\sigma/\sqrt{m}), \tag{C8}$$

which is only true if  $m \in \Omega(\alpha^2/\delta^2)$ .  $\square$

We therefore see that the number of samples needed to track the small changes in a posterior distribution that happens when the likelihood function becomes extremely flat. This condition is not sufficient because the actual components of the posterior mean may be shifted by a much smaller amount than the upper bounds used in the proof of [Theorem 1](#).

In contrast, exact Bayesian inference requires a number of bits that scales as  $O(\log(1/\delta))$  (assuming a fixed and discrete number of hypotheses). Thus exact Bayesian inference (or to a lesser extent particle filter methods) can be preferable in cases where the likelihood function is extremely flat. Such concerns can be somewhat be either be avoided entirely or batches of such experiments should be combined to produce a likelihood function that is much less flat and choosing an appropriate instrumental distribution to ensure that the success probability remains high.

#### Appendix D: Bayes Factors for Reset Rule

Though in the main body, we have chosen to present the reset step in terms of  $p$ -values for familiarity, we note that  $p$ -values are difficult to correctly interpret and can lead to misunderstandings [\[26, 27\]](#). As an alternative, one may prefer to use the Bayes factor to test whether the rejection filter has failed. For example, the rejection filter could fail due to a failure of the numerical approximation or because the eigenstate has been lost, as described in the main body. For a uniform prior over whether the rejection filter has failed, this reduces to the likelihood ratio test

$$L = \frac{\Pr(\text{result} = 1 | \text{prior wrong})}{\Pr(\text{result} = 1 | \text{prior correct})} \tag{D1a}$$

$$= \frac{1 - e^{-(\tau^2\sigma_{\text{reset}}^2/2\sigma^2 + \sigma\tau/T_2)} \cos(\tau(\mu - \mu_{\text{reset}})/\sigma)}{1 - e^{-\tau^2/2}} \tag{D1b}$$

where  $\mu_{\text{reset}}$  and  $\sigma_{\text{reset}}$  are the values of  $\mu$  and  $\sigma$  immediately following a reset. Using this test, the degree by which  $L > 1$  informs as to the degree to which we should prefer the model proposed by the reset rule. Again under the assumption of a uniform prior over the validity of the reset rule,

$$\Pr(\text{prior wrong} | \text{result} = 1) = L \Pr(\text{prior right} | \text{result} = 1). \tag{D2}$$

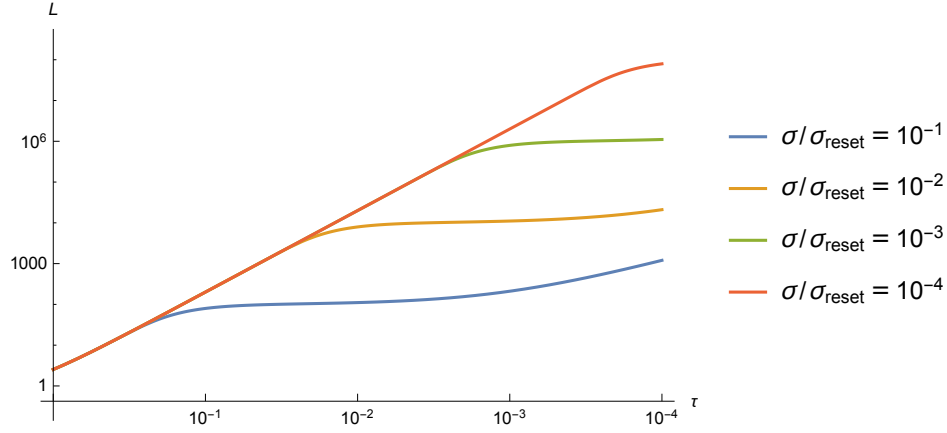


FIG. 9. Likelihood ratio test values for various settings of the parameter  $\tau$ , and for various prior variances  $\sigma/\sigma_{\text{reset}}$ . In this example, we suppose that  $\mu = \mu_{\text{reset}}$  and that  $T_2 = 100$ .

For instance, if the variance has been reduced by a factor of 100 from its initial value ( $\sigma = \sigma_{\text{reset}}/100$ ) and the current mean is correct ( $\mu = \mu_{\text{reset}}$ ), then assuming  $T_2 = 100$  and an initial standard deviation of  $\sigma_{\text{reset}} = \pi/\sqrt{3}$ ,  $L \approx 8000$  for a result of 1. That is, the initial prior is 8,000 times as probably correct as the current prior in this example.

Importantly, this example rests on the assumption that one take a uniform prior over whether the current prior is valid. This assumption corresponds to that which an impartial observer might take in evaluating whether the numerical approximations made by our rejection filter algorithm have failed for the observed data. That is, the reset rule proposed corresponds performing an intervention without relying on the full experimental data record. Choosing a threshold other than  $L = 1$  represents placing a different prior, as could be informed by observing the failure modalities of many runs of the rejection filter method. As demonstrated in Figure 9, because our reset rule resets with probability 1 if a 1 is observed, choosing  $\tau$  effectively sets the threshold for  $L$ . In practice, however, because  $\Pr(\text{result} = 1 | \text{prior correct}) \approx 0$ , the reset rule is only weakly dependent on the specific threshold one places on  $L$ .

### Appendix E: Pseudocode for Algorithms

In the main body we sketched the details of our phase estimation algorithm. Here we elaborate on this algorithm and discuss some of the subtle details needed to make the algorithm work. The first such subtlety stems from the fact that eigenphases are equivalent modulo  $2\pi$ . To see this problem, consider a Gaussian distribution centered at 0. If we take the outputs of the distribution in the branch  $[0, 2\pi]$  then we find that the mean of the distribution is  $\pi$  rather than 0. Since the support of the initial Gaussian may be small at  $\phi = \pi$ , such errors can be catastrophic during the inference procedure. This can be dealt by using the circular mean and by working with a wrapped normal distribution. This approach is discussed in Algorithm 1. For expedience, we eschew this approach and instead use a heuristic approach that does not require a large number of trigonometric calls, which can be expensive if the device performing the inference does not have native trigonometric calls.

The heuristic approach, described in Algorithm 2, uses rejection sampling and incremental methods to estimate the mean and standard deviation of the posterior distribution. If  $\sigma \ll 2\pi$  then the probability distribution is narrow and does not suffer from significant wrap around unless  $\mu \bmod 2\pi \approx 0$ . We address this by keeping track of each of the accepted  $\phi_j$  as well as  $\phi_j + \pi$ . If  $\sigma \ll 1$  then it is impossible that both distributions suffer from substantial wrap around. The arithmetic, rather than circular, mean and standard deviation are then computed for both using an incremental formula and the branch with the smallest variance is kept. If the branch that was shifted by  $\pi$  is chosen, then  $\pi$  is subtracted from the reported mean. The standard deviation does not need to be modified because it is invariant with respect to displacements of the mean.

While this approach is correct if  $\sigma \ll 1$ , it is only approximate if  $\sigma$  is on the order of 1. In such cases, computation of the circular mean is much better justified, however we find that using our heuristic approach continues to provide acceptable results while avoiding trigonometric calls that can be expensive in some contexts. An alternative approach to solving this problem is to begin each phase estimation run with a batch of random experiments, as per [9], before continuing to ensure that the posterior variance is small enough to neglect the wrap around effect.

The choice of the evolution time and the inversion angle strongly impacts the efficiency of the learning algorithm.

We provide below code for a modified version of the particle guess heuristic of [19]. As discussed in the main body, we expect that choosing  $M > T_2$  will typically lead to worse estimates of the eigenphase because the effects of decoherence overwhelm the information that can be gleaned from these long experiments. As a result, we modify the particle guess heuristic to never choose  $M > T_2$ . We formally state this procedure in [Algorithm 4](#).

---

**Algorithm 1** Bayes Update for RejF using Directional Statistics

---

**Input:** Prior mean and variance  $\mu, \sigma$ , measurement  $E$ , settings  $M, \theta$ , number of attempts  $m$ , scale  $\kappa_E$

```

▷ Initialize accumulators to 0.
 $(x_{\text{inc}}, y_{\text{inc}}, N_a) \leftarrow 0$ .
▷ Attempt each sample.
for  $i \in 1 \rightarrow m$  do
   $x \sim \frac{e^{-(\phi - \mu)^2 / 2\sigma^2}}{\sigma\sqrt{2\pi}}$ ,
  ▷ Accept or reject the new sample.
   $u \sim \text{Uniform}(0, 1)$ 
  if  $P(E|x) \geq \kappa_E u$  then
    ▷ Accumulate using the accepted sample using Cartesian coordinates.
     $x_{\text{inc}} \leftarrow x_{\text{inc}} + \cos(x)$ 
     $y_{\text{inc}} \leftarrow y_{\text{inc}} + \sin(x)$ 
     $N_a \leftarrow N_a + 1$ .
  end if
end for
 $x_{\text{inc}} \leftarrow x_{\text{inc}} / N_a$ 
 $y_{\text{inc}} \leftarrow y_{\text{inc}} / N_a$ 
▷ Return mean, variance of the posterior using accumulators.
 $\mu \leftarrow \text{Arg}(x_{\text{inc}} + iy_{\text{inc}})$ .
▷ Use circular standard deviation to estimate SD for wrapped Gaussian
 $\sigma = \sqrt{\ln \left( \frac{1}{\sqrt{x_{\text{inc}}^2 + y_{\text{inc}}^2}} \right)}$ 
return  $(\mu, \sigma)$ 

```

---

---

**Algorithm 2** Bayes Update for RejF

---

**Input:** Prior mean and variance  $\mu, \sigma$ , measurement  $E$ , settings  $M, \theta$ , number of attempts  $m$ , scale  $\kappa_E$

```

▷ Initialize accumulators to 0.
 $(\mu_{\text{inc}}, \mu'_{\text{inc}}, V_{\text{inc}}, V'_{\text{inc}}, N_a) \leftarrow 0$ 
▷ Attempt each sample.
for  $i \in 1 \rightarrow m$  do
  ▷ Draw a sample using each “cut” of the prior.
   $x \sim \frac{e^{-(\phi-\mu)^2/2\sigma^2}}{\sigma\sqrt{2\pi}},$ 
   $x \leftarrow x \bmod 2\pi.$ 
   $x' \leftarrow x + \pi \bmod 2\pi.$ 
  ▷ Accept or reject the new sample.
   $u \sim \text{Uniform}(0, 1)$ 
  if  $P(E|x) \geq \kappa_E u$  then
    ▷ Accumulate using the accepted sample w/ each “cut.”
     $\mu_{\text{inc}} \leftarrow \mu_{\text{inc}} + x$ 
     $V_{\text{inc}} \leftarrow V_{\text{inc}} + x^2$ 
     $V'_{\text{inc}} \leftarrow V'_{\text{inc}} + x'^2$ 
     $N_a \leftarrow N_a + 1.$ 
  end if
end for
▷ Return mean, variance of the posterior using accumulators.
 $\mu' \leftarrow \mu_{\text{inc}}/N_a$ 
 $\sigma' \leftarrow \min\left(\sqrt{\frac{1}{N_a-1}}(V_{\text{inc}} - \mu_{\text{inc}}^2), \sqrt{\frac{1}{N_a-1}}(V'_{\text{inc}} - \mu'^2_{\text{inc}})\right)$ 
return  $(\mu', \sigma')$ 

```

---



---

**Algorithm 3** Restarting algorithm

---

**Input:** Prior RejF state, records of all previous models found in the phase estimation algorithm  $\mu, \vec{\sigma}$ , initial standard deviation  $\sigma_{\text{init}}, M, T_2$ , a counter CNT,  $\Gamma$  and  $\tau$ .

**Output:** CNT,  $\sigma$

```

function RESTART( $\mu, \vec{\sigma}, \sigma_{\text{init}}, M, T_2, \text{CNT}, \Gamma, \tau$ )
   $D \leftarrow$  derivative of  $\log \sigma.$ 
  if  $D \geq \Gamma$  and  $\text{CNT} < 5$  or  $\text{rand}() > \exp(-M/T_2)$  then
    Perform experiment with  $M = \tau/\sigma$  and  $\theta = \mu.$ 
    if Outcome is 0 then
      CNT  $\leftarrow$  CNT + 1.
      return CNT,  $\sigma.$ 
    else
      CNT  $\leftarrow$  0
       $\sigma \leftarrow \sigma_{\text{init}}$ 
      return CNT,  $\sigma$ 
    end if
  else
    CNT  $\leftarrow$  CNT + 1
    return CNT,  $\sigma.$ 
  end if
end function

```

▷ Checks to see if the eigenstate is suspect.  
 ▷ Test concludes state estimate is valid  
 ▷ Test concludes state estimate is invalid  
 ▷ Does not restart if state is not suspect

---

---

**Algorithm 4** PGH for decoherent phase estimation using RejF

---

**Input:** Prior RejF state  $\mu, \Sigma$ . Resampling kernel  $F$ .

**Output:** An experiment  $(M, \theta)$ .

**function** PGH<sub>RejF</sub> ( $\mu, \Sigma, T_2$ )

$M \leftarrow 1.25/\sqrt{\text{Tr}(\Sigma)}$

**if**  $M \geq T_2$  **then**

$M \sim f(x; 1/T_2)$

$\triangleright$  Draw  $M$  from an exponential distribution with mean  $T_2$ .

**end if**

$(-\theta/M) \sim F(\mu, \Sigma)$

**return**  $(M, \theta)$ .

**end function**

---