Universidade do Minho
School of Engineering

Alexandra Ramôa Alves

Noise resilient quantum amplitude estimation

March, 2023

Universidade do Minho
School of Engineering

Alexandra Ramôa Alves

Noise resilient quantum amplitude estimation

Doctorate Thesis Proposal
Doctorate in Informatics

Work developed under the supervision of:

Luís Paulo Santos
Ernesto Galvão

March, 2023

Resumo

Estimativa de amplitude quântica resiliente ao ruído

A estimativa de amplitude quântica é a tarefa de estimar a probabilidade de medir um dado estado quântico que integra uma sobreposição. O seu análogo clássico consistiria em determinar o parâmetro subjacente a uma distribuição de Bernoulli. Neste último caso, a melhor solução é pouco criativa, baseando-se em recolher amostras e calcular a sua média. Já no caso quântico, graças à liberdade adicional oferecida pela mecânica quântica, existem estratégias mais interessantes. Para além do mais, tais estratégias são capazes de obter uma vantagem de complexidade relativamente ao caso clássico - no melhor caso, uma vantagem quadrática.

Esta vantagem quântica é semelhante à conseguida na procura numa base de dados sem estrutura. Apesar de modesta quando comparada com a vantagem exponencial do celebrado algoritmo de fatorização quântica, o seu potencial é mais amplo. Em particular, a integração de Monte Carlo pode ser formulada como um problema de estimativa de amplitude, e portanto beneficiar do uso de recursos quânticos tal como esta. Dadas por um lado as extensas aplicações deste método de integração, e por outro o seu custo computacional limitador, este é um panorama tentador.

No entanto, tal como acontece com outras proezas quânticas promissoras, esta encontra-se ainda limitada ao domínio da teoria. As razões para tal são duas. Uma são as limitações dos dispositivos quânticos da atualidade; e a outra é a natureza intransigente dos algoritmos que os poderiam usar. É portanto de esperar que o caminho mais curto para demonstrações experimentais passe por um esforço concertado que aborde o problema destes dois ângulos.

Considerando isto, o presente trabalho concentra-se em métodos de estimativa de amplitude que sejam mais lenientes perante falhas do equipamento físico, e por conseguinte menos suscetíveis a ruído e erros experimentais. Nesse âmbito, passamos em revista os algoritmos existentes, e analisámo-los teórica e numericamente, com ênfase na tolerância a falhas. Isto inclui simulações tanto com como sem ruído. Por fim, testamos um algoritmo adaptativo Bayesiano assintoticamente correto, e propomos direções para trabalho futuro.

Abstract

Noise resilient quantum amplitude estimation

Quantum amplitude estimation is the task of estimating the measurement probability of a given quantum state partaking in a superposition. Its classical counterpart would consist of learning the underlying parameter in a Bernoulli distribution. In the latter case, the best performing solution is rather unimaginative, relying on drawing and averaging samples. In the former, owing to the extra freedom afforded by a quantum-mechanical treatment, more spirited strategies can be found. What is more, they can offer up to a quadratic speed-up relative to the classical case.

This quantum-powered advantage is akin to that achieved in unstructured database search. Albeit modest as compared to the exponential advantage of the celebrated quantum factoring algorithm, its potential reach is wider-ranging. In particular, Monte Carlo integration can be formulated as an amplitude estimation problem, and thus benefit from the same quantum enhancement as the latter. Given on one hand the far-reaching applicability of this integration method, and on the other its constricting computational cost, this is a rather alluring prospect.

However, as with other promising quantum feats, this one is yet to break free from the realm of theory. The reasons for that are twofold. One are the limitations of the currently available quantum devices; and the other is the unaccommodating nature of the algorithms meant to use them. It is then to expect that the fastest path towards experimental demonstrations is a concerted effort tackling the problem from these two angles.

With this in mind, the present work focuses on forms of amplitude estimation that are lenient with respect to hardware flaws, and thereby less susceptible to noise and experimental errors. We review existing quantum amplitude estimation algorithms and analyse them from theoretical and numerical perspectives, with an emphasis on fault resilience. This includes simulations both with and without noise. We then attempt an asymptotically correct adaptive Bayesian algorithm, and point directions for future work.

Contents

Appendices

List of Figures

Acronyms

AES	(quantum) amplitude estimation, simplified *(pp. 29–31, 37, 52)*
EIS	exponentially incremental sequence *(p. 52)*
FAE	faster (quantum) amplitude estimation *(pp. 36, 37)*
HL	Heisenberg limit *(pp. 43, 46, 50–52, 56)*
IAE	iterative (quantum) amplitude estimation *(p. 37)*
LIS	linearly incremental sequence *(pp. 27, 52)*
M-IAE	modified (quantum) iterative amplitude estimation *(pp. 36, 37)*
MLAE	maximum likelihood (quantum) amplitude estimation *(pp. 25–28, 37, 52)*
MLE	maximum likelihood estimation *(pp. 23, 26, 50, 51)*
NISQ	noisy intermediate scale quantum *(pp. 3, 8, 24, 63)*
PLAE	power law (quantum) amplitude estimation *(pp. 28, 29)*
QAA	quantum amplitude amplification *(pp. 1, 26, 37)*
QAE	quantum amplitude estimation *(pp. 1, 2, 8, 20, 22, 24, 26, 29, 30, 34, 36, 37, 41, 42, 50, 60, 61, 63, 73)*
QAOA	quantum approximate optimization algorithm *(p. 4)*
QFT	quantum Fourier transform *(pp. 6–8, 18, 26, 29, 55, 72, 73)*
QPAE	QoPrime (quantum) amplitude estimation *(pp. 28, 29)*

QPE	quantum phase estimation *(pp. 5, 6, 10, 18–21, 26, 30, 37, 72, 73)*
RAE	robust (quantum) amplitude estimation *(pp. 60, 61)*
root mean squared error	root mean squared error *(pp. 44, 49)*
SAE	simpler (quantum) amplitude estimation *(pp. 34, 37, 53)*
SQL	standard quantum limit *(p. 50)*
V-MLAE	variational maximum likelihood (quantum) amplitude estimation *(pp. 27, 37)*
VAE	variational (quantum) amplitude estimation *(p. 27)*
VQE	variational quantum eigensolver *(pp. 3, 4, 61)*

1

Introduction

1.1 The quantum amplitude estimation problem

This section means to describe what quantum amplitude estimation (QAE) consists of, without diving into mathematical details. The main reference is [4].

The QAE problem arises naturally in the same context as that of **Grover's quantum searching algorithm** [23]. There is a database containing many items, a select few of which have a specific property that we can easily identify upon seeing it. One may imagine a storehouse with many identical boxes; most of them are empty, but some are treasure chests.

Grover's algorithm solves the task of finding one of these special items by (figuratively) opening the fewest possible boxes. Classically, given the lack of structure in the database, our best strategy is to go through the items one by one until we find the treasure. In contrast, the use of quantum resources enables a faster path. The key ingredient is our ability to recognize special items when we see them. Armed with such an ability, we can construct a valuable quantum operator that acts on a superposition of special and non-special items by increasing the amplitudes of the former and decreasing those of the latter.

As such, if we apply this operator the correct number of times and subsequently sample an item from the superposition, this item is more likely to be special than before. This impressive feat is due to a process called quantum amplitude amplification (QAA), which is made possible by quantum interference. The aforementioned ability to recognize special items can be used to treat them differently, in such a way that the interference builds them up while having a destructive effect on others.

Considering the cost of the amplification and how much it improves our odds, the quantum algorithm offers a **quadratic speed up** relative to the classical brute-force approach (which is the optimal approach under the stated conditions). It has been demonstrated that this is the best achievable speed-up for this task: Grover's algorithm cannot be improved upon [67].

It should be mentioned that the action of the amplification operator is cyclic, so applying it too many times could be counterproductive. If we know how many special items there are, we can find the best number of operator applications. (In cases where the number of marked items is not known, there exist workarounds [4].)

Depending on the ratio of special items to total items, we may or may not be able to achieve a 100% success rate through amplification. If not, we can alter the originally proposed operator to correct that [4]. Alternatively, we can accept the non-null probability of having to try again; it does not affect the quadratic advantage in the average case [4].

In **quantum amplitude estimation**, the focus shifts from *finding* the number of special items to *counting* how many they are. We can think of a pirate who owns the previously mentioned storehouse and wants to quantify his financial assets, which consist of an unknown amount of treasure chests. This can be further generalized to consider non-integer numbers of items, as well as items of varying importance.

Interestingly, this very different task can be aided by the same tool as the previous one: amplitude amplification. What is more, this tool brings the same quadratic speed-up.

In the case of quantum amplitude estimation, the perk of working in the quantum domain is that we can make measurements at different levels of amplification; whereas in the classical case, we are not granted any such luxury. In other words, in the quantum case we can sample from a binomial distribution with a controllable parameter, whereas in the classical one we are restricted to a fixed probability of success. As we will see, this extra degree of freedom allows us to learn the amplitude more efficiently.

1.2 Applications of amplitude estimation

In the previous section, we presented the problem of amplitude estimation abstractly. We would now like to introduce some contexts in which it represents a pertinent task, so that solving it efficiently is of practical interest.

Perhaps the most prominent domain of application is Monte Carlo simulation. This is an umbrella term for stochastic methods that generate samples distributed according to a probability distribution of choice, which can then be used to numerically estimate expectation values of functions. Differently put, by averaging over the samples, we can integrate with respect to their underlying probability measure.

Interestingly, this method can be formulated as an amplitude estimation task. As a matter of fact, the speed up provided by the quantum routine for that task transfers in full to the task of computing a Monte Carlo estimate of an expected value.

This is quite promising, as Monte Carlo methods are a very powerful tool - and one which is widely used across a multitude of fields, such as finance, engineering, and science. Their main downside being a high computational cost [5], a prospective speed up is quite alluring.

While the quadratic quantum advantage seen in QAE is rather modest compared to e.g. the exponential speed up afforded by quantum resources in Shor's algorithm, the vastness and relevance of its applications make it quite attractive. Unsurprisingly, the prospect of such a far-reaching improvement has sparked some interest in this quantum routine, especially as applied to finance [11], where QAE can aid in tasks like risk analysis [66, 20] and option pricing [50, 40, 55, 20].

The QAE routine can also find utility within the realm of quantum mechanics - namely in quantum metrology, where it allows for surpassing the standard quantum limit. This limit describes the best achievable performance for a classical estimation algorithm, which seeks to estimate a quantity based on experimental data. A good performance maximizes the precision while minimizing the resources spent. The use of quantum resources allows for working in the quantum-enhanced estimation regime, which is at best Heisenberg-limited - bringing up to a quadratic improvement in the learning rate.

Clearly, the most straightforward estimation task to benefit from this enhancement is the estimation of the amplitude itself [4]; but similar improvements apply to the estimation of other quantities. An example are the expectation values of observables [34], which in turn can be useful in various scenarios, such as in the simulation of chemical reactions [30] or in the variational quantum eigensolver (VQE) [31, 29].

Quantum amplitude estimation can also bring a complexity advantage in multiple machine learning algorithms, such as deep learning [64], classification [63], clustering [63, 32], and reinforcement learning [65]. In all of these, amplitude estimation is used to approximate quantities of interest faster than would be possible classically; for instance, for evaluating gradients, centroid distances, or policies.

1.3 Noise resilient algorithms

In the previous sections, we focused on the last three words of this work's title. This one is dedicated to the first two: it aims to discuss noise resilient algorithm design.

This matter is deemed pertinent due to our living in what is called the noisy intermediate scale quantum (NISQ) era, an epithet describing quantum devices that are noise-susceptible and not very large (in terms of the number of qubits) [48]. These two attributes are correlated, as a limited size makes error correction unfeasible; and the stronger the noise, the more unfeasible it becomes.

We must then work with quantum devices that do not have as many qubits as we would like, and suffer from various evils that endanger information processing and extraction. In particular, these devices are plagued by decoherence, which limits the time a computation can take while still producing useful results. Should this time be crossed, the information will be erased, leaving essentially noise at the output. Yet other sources of error exist, such as random bit and phase flips, cross-talk, and gate miscalibration.

Due to these problems, the most impressive quantum algorithms have only been implemented to solve very small, proof-of-principle instances [53, 16]. This includes Shor's prime factoring algorithm [52], which offers an exponential speed-up relative to the best performing classical algorithm, and Grover's unstructured database search algorithm, which offers a more modest - but also more versatile - quadratic speed-up [23].

Despite the theoretical soundness of these algorithms, their results have never been demonstrated experimentally for problem instances of considerable size. This means that a practical quantum advantage has never been observed: it awaits fault-tolerant quantum devices that some argue will never arrive. Even if they do eventually become reality, it is a distant reality, and immense progress in engineering and error correction separates us from it.

In the meantime, on the algorithmic side, accommodations can be made or attempted to extract some advantage from the faulty machines we already have. The benefits of evolving theory alongside practice are manifold. It allows us to test evolving technology; encourages and strengthens experimental efforts; fosters the development of quantum information theory; and keeps the interest in quantum computing alive.

The price to pay is having to work with the limitations of the current technology. Algorithms must not only economize qubits, circuit depth, and complex operations, but also avoid the overreliance on exactness that characterizes fault-tolerant algorithms. A common approach is to switch from the canonical strategies based on a single long circuit, to others using several shorter circuits. The latter are often complemented with classical processing and adaptivity, which orchestrate the various circuits and compile their results. This switch usually implies changing from instant result extraction on read-out to sequential knowledge refinements. In such alternative arrangements, the circuits are often more amenable to imperfect hardware, and so too is the processing. Emblematic examples are the VQE [45] and the quantum approximate optimization algorithm (QAOA) [13] algorithms.

It is then no surprise that similar approaches have been developed for quantum amplitude estimation. Such approaches are what this project means to explore and expand upon.

1.4 A bird's eye view

This section will briefly overview the existing literature on quantum amplitude estimation, without getting into technical details. This algorithm was first presented in 2000 by Brassard and co-authors [4] as a blend of Grover's [23] and Shor's [52] routines, perhaps the two best-known quantum algorithms.

The framework is set by Grover's algorithm, which conducts a quantum-enhanced search on an unstructured database. It can find a single distinguished item quadratically faster than the best classical algorithm. This distinguished item is marked somehow; thus, we can easily tell whether an item is *the* item. However, *finding* the one marked item is not as simple. Our goal is to do so while checking the fewest items possible.

Once again, we invoke the storehouse metaphor. This storehouse is full of boxes, all but one of which are empty; the remaining one contains a treasure, which we are interesting in finding. The standard strategy would be to go through all the items until we discover the special one; meaning, we'd peek into each box until we see the treasure. This could mean checking all of them in the worst case.

Grover's approach provides an alternative path. It uses quantum states to represent tentative items, whereby they can exist in a superposition. The starting point is then an initial superposition state; in the absence of a priori information, this can be an equally weighted superposition of all possible items. If we have some previous knowledge, we can use it; for example, if we know in advance that some specific items are not marked, we can exclude those. More generally, should we have any motivation to check the boxes in a non-arbitrary (or at least not *entirely* arbitrary) order, that search heuristic can too be incorporated into the quantum approach.

Additionally, the protocol requires access to a quantum phase oracle: an operator which, upon acting on a quantum circuit, identifies the distinguished item (codified as a quantum state) by marking it with a phase. Having the ability to physically implement such an oracle, the speed-up can be achieved by means of a unitary operator, hereby called the Grover operator, which is tailored to both the oracle and the initial state. This operator is then repeatedly applied to the latter.

Up to a certain point, which is known, this process boosts amplitude of the marked state, and thus the probability of finding the marked state upon measuring the circuit. It is only after continuously enhancing this success probability - bringing it as close as possible to 100% - that we measure the circuit. A thus constructed measurement is likely to immediately yield the item we are interested in finding.

That this can be achieved with a single measurement may sound a little too good to be true. What is the catch? First off, it should be stated that requiring a single measurement is not equivalent to opening a single box, as the amplification itself requires operations that amount to box peeking. But even if we quantify this resource correctly, we have a quadratic advantage over the basic box-checking solution.

How about this *high probability* - does it hide something? Not quite, for two reasons. First, the probability can always be made at least as high as 50%, meaning that any given attempt is more likely than not to succeed - and if not, the process can be repeated until one does. Remember that we can easily check whether we succeeded: it is straightforward to verify whether there's a treasure in a box. If not, we repeat the amplification and measurement operations as many times as necessary until we succeed. These repetitions do not affect the quadratic advantage in the average case.

Even still, should the randomness make us uneasy, we can de-randomize the procedure. Such an idea was presented in [4], which offers two possible paths to achieve it: by re-framing the problem, or by customizing the operator. Again, these modifications preserve the quadratic speed-up.

This concludes our discussion of the Grover algorithm. In [4], the authors generalize it further by considering the possibility of there being multiple marked states, which changes how many Grover iterations are necessary. Subsequently, they consider a different question altogether, which we are particularly interested in. It can be formulated as: how to count the number of marked states?

This task is generally called *quantum counting*. Again, the goal is to keep the number of Grover iterations to a minimum. In this case, the initial state is a uniform superposition; and ultimately, we want to find the probability that measuring it yields a marked state. Because the total number of items is known, this probability is enough to tell us how many of them are marked.

In the classical case, we would open all of the boxes, keeping track of how many have treasures. Alternatively, if we were content with an approximate solution, we could consider only a fraction of the boxes, and assume the proportion of empty ones found therein approximates the global one.

As for the quantum case, it can be aided by Shor's algorithm, or a component thereof. It can be shown that the fraction of marked items can be deduced from the phases of Grover operator's eigenvalues. Thus, the quantum counting problem amounts to the problem of estimating these eigenphases; this can be realized by quantum phase estimation (QPE) [33], which is precisely the backbone of Shor's algorithm

- hence the description of quantum counting as a Grover-Shor mixture. For a short discussion of QPE, we refer to appendix A.

To generalize this to amplitude estimation, it suffices to consider non-integer numbers of marked items. While nonsensical in the database problem, this is straightforward when thinking in terms of the quantum operations. Should the quantum oracle mark a basis state whose coefficient is complex-valued, we can use the same resources as before to learn its absolute value. This only differs in that there ceases to be an interpretation in terms of a discrete database.

In this case, the corresponding classical problem would be to estimate the parameter of a Bernoulli distribution by sampling from it. A satisfactory approximation could be achieved by averaging the number of 1 outcomes over a sufficiently large number of trials.

With this we have laid out the quantum amplitude estimation problem, as well as the canonical way of solving it. Unfortunately, the latter cannot yet be brought to fruition. This is due on one hand to the complexity of the required quantum circuits, and on the other to the limitations of today's quantum devices. As for the former, the main culprit is the QPE routine. It relies on a long sequence of controlled applications of the Grover operator, followed by a quantum Fourier transform (QFT) - which consists of yet another long sequence of controlled operations.

For any problem of relevant size, this is incompatible with the current technology, which is plagued by noise. In particular, quantum devices suffer from a phenomenon called *decoherence*, which results in the progressive erasure of information. If the computation takes too long, all information will be gone at the time of termination, leaving no meaningful results. Furthermore, the implementation of controlled operations through entangling gates is a problem of its own, as these gates are particularly susceptible to noise.

In this context, attempts have been made to soften the demands placed on the quantum hardware, by designing quantum algorithms that are more forgiving with regards to the aforementioned experimental imperfections. This mostly means using shallower circuits, and simpler operations.

There is a very simple way to perform amplitude estimation with constant depth circuits, and without any controlled Grover iterations at all. We simply create a uniform superposition, measure immediately, and check whether the measurement outcome is marked. After repeating this procedure for many trials, we calculate the fraction of times the output was a marked item. This average success rate is precisely an estimate of the amplitude we mean to find.

Unfortunately, this extremely hardware-efficient approach is equivalent to the classical one, bringing no advantage whatsoever; except for the touch of whimsy in the implementation, which is (arguably) hardly worth the trouble.

With this in mind, we can reframe the objective as: how to soften the demands placed by amplitude estimation on the quantum hardware, *while preserving the quantum advantage*? What we want is an algorithm that retains as much of the speed up as possible, while being hardware-friendly.

A good start is to scrap the QFT. In Shor's algorithm, this lavish transformation is associated with an exponential speed up; so it seems rather unfair that it would be indispensable for a mere quadratic one. This observation raised the question of whether the speed-up achieved by [4] could forgo the Fourier transform.

In 2019, this question was answered affirmatively by Aaranson and Rall [1], who replaced the QFT by a sequential scheme relying on direct amplitude amplification - that is, on Grover-type circuits, where the only variable is the number of Grover iterations.

Even though this accomplished the intended speed up, a large constant factor was involved; meaning that even though the uncertainty in the amplitude shrinks as fast as desired, it starts out unfavorably. As such, open questions remained: is it possible to improve this further? And can the asymptotic behavior be safeguarded while doing so?

In the same year, other authors had set forth their own alternatives, trading off rigor for practicality. In [56], Suzuki and co-authors proposed a maximum likelihood approach. Their strategy relies on heuristic sequences of simple Grover circuits, and infers the amplitude based on the measurement data extracted from them. The authors prove lower, but not upper, bounds for the estimation error. However, a numerical analysis shows good performance.

Two years later, [47] considered reworking this maximum likelihood algorithm from another perspective, by periodically replacing chunks of Grover iterations with variational approximations in order to reduce circuit depth. They demonstrate interesting numerical results, despite incurring a cost overhead due to the variational circuit optimization. In the same year, an even more interesting proposal came from [19], which reworked the scheme of [56] to cover the ground between the classical and quantum approaches - the goal being to mindfully exploit the limited *quantumness* of near-term devices. On top of that, the authors developed another algorithm achieving the same feat while offering stronger formal guarantees. Both approaches demonstrate robust numerical performances, and resist well against decoherence.

Shortly after [56], [61] introduced a straightforward approach based on Hadamard tests, which they called simpler quantum counting. After several executions, the model for the outcome distribution is inverted to obtain the parameter of interest. However, neither the theoretical nor the numerical analyses are very robust, in that they don't directly address the performance metrics of interest.

Not much later, [22] came up with an algorithm that combined formal rigor, a solid numerical performance, and a modest cost offset - a feat none other had achieved until then. Although it couldn't attain the ideal asymptotic complexity, it came quite close, with only a double-logarithmic factor separating the two. What is more, the experiments demonstrated its competitiveness, which was unmatched by any other algorithm.

This remarkable algorithm drew the attention of other authors, prompting modified versions. In particular, [17] enhanced it through a rearrangement of its failure probabilities across iterations - managing to shave off the unwanted logarithmic factor to get an optimal asymptotic performance. This development further consolidated the significance of this algorithm as a both rigorous and practical approach. Yet its noise-obliviousness renders it impractical for near-term use, where it loses against more heedful strategies such as those of [19].

Another algorithm was later proposed in [42]. Again it falls short of achieving Heisenberg scaling, but again it comes close, the difference being yet again a double-logarithmic factor. It relies on straightforward inversions of circuit until they are barred by redundancy; at which point it changes into more involved inversions requiring additional measurements for disambiguation. As in [22], numerical tests show quite satisfactory results.

In 2021, [59] advanced an approach set apart by its versatility. Instead of relying (as most others) on rigid schedules and meticulous calculations underpinned by a rigorously crafted analytical backbone, it relied on a less stiff, but no less powerful, framework based on Bayesian inference [35]. Not only is this framework capable of noise mitigation [31], these capabilities can be combined with or enhanced by complementary techniques [7]. The richness of this take immediately opens a multitude of interesting paths to pursue, some of which have brought QAE closer to practical applications [29].

1.5 Research questions

Having presented a bird's eye view of pre-existing literature, it is time to discuss open questions.

The overarching goal is clear: to achieve optimal asymptotic performance (Heisenberg scaling) with a reasonable cost offset. The latter caveat intends to safeguard the feasibility of the algorithm beyond its *efficiency* in terms of learning rate. Put differently, we want to attain a quadratic speed-up in the error reduction rate relative to the classical case, without starting at too unfavorable a position (which would risk canceling out the speed-up for practical problem sizes).

Some of the algorithms described in the previous section already achieve this goal, or come satisfactorily close to it. As a matter of fact, should this be enough, we could have stayed by the canonical algorithm of [3] to begin with. But we want more than that: we want to be able to achieve this *with imperfect technology*.

Such is the motivation behind the flurry of alternative algorithms the past years have witnessed. The ambition is to preserve as much of the ideal scaling as possible, while easing the requirements on the quantum hardware. This is usually interpreted as reducing the number of qubits, the depth of the circuits, and the complexity of the operations.

Accordingly, the first preoccupation should be to eliminate the QFT and controlled Grover iterations; and the second to remove the controlled Grover iterations. Addressing these points should make the algorithm more forgiving, and bring it a bit closer to reality.

Even still, it does not bring it close enough to reality that we can implement it *today* - and as long as that's true, there's room for improvement. In particular, algorithm design can be adapted still more drastically to consider the imperfect technology it must work with. The third concern should then be to increase the *noise resilience* of the approaches.

Even if the developed algorithms are "NISQ-friendly" - in the sense that their physical implementation is a less onerous task than the original algorithm of [4] -, most of them assume fault tolerant computations. In other words, the circuits they employ are easier on the hardware by construction, but are still expected to yield perfect results upon execution.

Not only does this underpin most or all proofs of correctness, it is also assumed when constructing measurement schedules. More specifically, to refine knowledge, algorithms usually rely on circuits of progressively increasing depth. Clearly, if this carries on indefinitely, the circuits will become too deep, to the point little to no information can be extracted from them. Once they exceed the device's coherence time, which in practice is always finite, we will be measuring classical noise at the output.

In this case, we would benefit more from a shallower circuit than from the one carefully picked by the algorithm schedule: the former can produce some new insight, whereas the latter cannot. This type of realistic consideration is rarely ever considered in the algorithm design. Some algorithms even require that the amplitude can be amplified to near unit values, which can render them entirely unfeasible for small amplitudes and modest coherence times. In this sense, it would be of interest to customize algorithms to hardware specifications, namely their average coherence times.

Another aspect that makes algorithms very noise-susceptible is the assumption of noiselessness (apart from shot noise), which leaves them defenseless in the face of any other sources of randomness such as experimental imperfections. This overreliance on exact outcomes makes them incapable of recovering from aberrant measurements. We can think of how e.g. iterative phase estimation (appendix A) is a shallower circuit-based alternative to sequential phase estimation, making it more hardware-friendly; yet it is clearly very fault-susceptible, as its results are severely compromised the moment an experimental error occurs in a single measurement.

With this in mind, another interesting direction for research is the adaptation of the processing techniques to accommodate fortuitous errors, or at least not be completely undermined by them.

Apart from this topic, some others merit further work. For instance, the approach of [56] is - unlike any other - entirely parallelizable, but does not achieve Heisenberg scaling. Would it be possible to conciliate these two qualities? The fact that nearly all quantum algorithms are parallel across the (rather large) number of measurement repetitions makes this question less pressing than it would otherwise be, but it is nevertheless relevant.

Yet one more pertinent subject is the classical processing. Most of these algorithms trade off quantum resources for classical ones. How much can we reduce the latter? Often, the classical processing must be realized in real time, sandwiched between one quantum circuit execution and the next; affected by the former, and effecting the latter. How much can this *online* processing be reduced? In other words, how necessary is adaptivity as a resource?

Finally, some topics of interest are application-specific; of those, we are particularly interested in the ones pertaining to numerical integration. Efficiently preparing quantum states according to the intended probability measure is perhaps the most notable challenge [11, 26], as the cost of loading the probability distributions may cancel out the quantum advantage [25].

This concludes the most relevant open questions on the topic. This work will focus on the ones regarding noise, and on Monte Carlo integration as an end-goal. The driving question can be phrased as: how much of the quantum advantage can be preserved in the presence of experimental shortcomings?

Under ”shortcomings” we include not only limitations, but also *imperfections*.

1.6 Related topics

In this section we intend to dedicate a few words to topics related to quantum amplitude estimation.

The two most glaring ones have already been mentioned: Grover's algorithm, and quantum counting. Amplitude estimation shares its framework with the former, and can be seen as a generalization of the latter.

Apart from these, some others stand out. For instance, the algorithm is closely related to QPE. As a matter of fact, in its original version, it consists of QPE applied to a particular target: the Grover operator. This also links it to Shor's algorithm, the most celebrated application of the QPE routine. This link led its creators to describe it as a mixture of Grover's and Shor's routines [4].

Does the problem of amplitude estimation then reduce to that of QPE, being just a particular case? And would attempts to modify it amount to attempts to modify the more general QPE routine? Expectably, this is not quite true. The Grover operator behaves in a very particular way, this special behaviour being what underpins the quantum search algorithm.

More directly, its remarkable property is that when acting on its corresponding initial state, it amplifies the amplitude according to a known scheme. Phase estimation is just one way of profiting from the richness of this setting - one which is especially elegant, but also especially tricky to implement. As we have discussed, amplitude estimation can dispense with it while still making the most of the Grover operator. (For a slightly more detailed discussion, refer to appendix A.)

In short, even if QPE solves the quantum amplitude estimation problem, it is not the *only* possible solution. (Here by quantum phase estimation we mean the quantum phase estimation *routine*; technically we always do *estimate a phase*, just by other means.)

Once we remove QPE from the picture, what we have is a reproducible and controllable system we want to characterize. To *characterize* is to find the amplitude (or some proxy parameter), and the *control* is the number of Grover iterations. This can be seen as a quantum characterization task; more specifically, as a quantum parameter estimation problem. These are topics of intense research in the field of quantum metrology.

In the case of amplitude estimation, the *control* determines how much the amplitude is boosted before any given measurement; the amplified amplitude evolves according to a squared-sinusoidal function of the number of Grover iterations applied between initializing and measuring. We can evaluate this function at discrete points, measuring their associated amplitude.

The setting we just described is commonly found in the context of precession dynamics associated with superconducting qubits, barring one difference: the function evaluations are continuous. Three well-known phenomena that illustrate this are Larmor [38], Rabi [39] and Ramsey [49] oscillations.

Larmor oscillations occur when a magnetic field is applied to a system with a magnetic spin. This causes the spin to oscillate around the field's direction, which is typically defined as the z direction. Furthermore, the oscillation frequency, also called the Larmor frequency, is proportional to the magnitude of the applied field. This can be used to physically implement qubits. In this context, spin measurements can be used to characterize the Larmor frequency, much like Grover-style measurements can be used to estimate amplitudes.

If we add a particular field contribution to the Larmor framework, we will observe what is called Rabi flopping. This contribution must be perpendicular to the direction of the Larmor polarization, and oscillate around it. If this is satisfied, Rabi oscillations occur: the populations of the qubit's two energy levels undergo a cyclic evolution.

This phenomenon can be used to implement quantum gates. For that, one must work with resonant radiation, i.e. the driving Rabi frequency must match the Larmor frequency of the qubit. The Bloch vector of a qubit can then be controlled via appropriate choices of the radiation's direction and duration.

On the other hand, a Rabi experiment can be used to learn the resonant frequency of a qubit - which is crucial information for gate calibration, according to the previous paragraph. This can be achieved by acting on the quantum system with radiation of known parameters, and analysing its response. Again, this is analogous to how we learn an amplitude by acting on a quantum circuit with a well-defined operator and analysing its output.

For this particular purpose, a modified version of the Rabi technique exists, and it relies on the so-called *Ramsey* oscillations. They are often of use in high precision metrology, in what is termed Ramsey interferometry. The technical details are outside of the scope of this work; but for the sake of completeness, we will say that it manages similar dynamics to Rabi's while being less susceptible to experimental issues.

Additionally, Ramsey interferometry can be applied to magnetic field sensing, enabling the realization of nanoscale magnetometers based on nitrogen-vacancy centers [58, 51].

To conclude, we add that in the field of photonics, Mach-Zehnder interferometry too is equivalent to all of these phenomena: estimating the relative phase shift between the two arms of the interferometer is equivalent to the above-mentioned problems [46].

The key takeaway is that *all* of these dynamics admit squared-sinusoidal descriptions, just as Grover circuits do. The difference is that in the latter case, the *time* has discrete units, which correspond to the number of times the Grover operator acts on the initial state. In contrast, qubit oscillations are a continuous phenomenon: we can measure them at any given time. This being essentially the only distinction between the two problems, techniques applied to one of them may transfer - partially or in full - to the other.

Lastly, we would like to mention Quantum Monte Carlo [44]. This algorithm may seem to be associated with numerical integration, which is a highlighted application of quantum amplitude estimation. However, it does not refer to quantum-enhanced Monte Carlo integration, but rather to a classical Monte Carlo algorithm used to study quantum systems. To avoid this ambiguity, we will refer to quantum-enhanced Monte Carlo as such, or alternatively as quantum Monte Carlo *integration*.

1.7 Document structure

This document is organized as follows. Chapter 2 offers a more in depth overview (as compared to section 1.4) of pre-existing literature on quantum amplitude estimation, including ”near term” algorithms. It also includes a short description of how numerical integration can be formulated as an amplitude estimation problem.

After that, chapter 3 describes the results obtained thus far in the scope of this work. For that purpose, we begin by introducing the concepts and metrics that are necessary for assessing algorithm performance. Having done that, we present the results obtained by numerically simulating a selection of previously proposed algorithms; first under ideal circumstances, and then under the presence of decoherence, a ubiquitous source of noise in quantum devices. This chapter ends with the introduction of a Bayesian approach, akin to others proposed in the literature but employing different numerical methods.

Finally, in chapter 4, we conclude by pointing out directions for future work, namely those laid out by the Bayesian framework. We end with a provisional timeline for the execution of this project.

2

Literature overview

This chapter intends to review the existing literature on quantum amplitude estimation. It is a more in-depth version of section 1.4.

2.1 Quantum searching: Grover's algorithm

The quantum amplitude estimation problem occurs naturally in the context of Grover search, so we must start by explaining the latter. This section is devoted to that objective.

The problem is as follows. There exist N items, each represented by a binary string x - or in the quantum case, a computational basis state $| \, x \, \rangle$ on $n = \log_2(N)$ qubits, belonging to a Hilbert space of dimension N. Of those items, exactly one is a solution to some specified problem.

It is given by the problem statement we can easily recognize (but not find) a solution. Mathematically, we can describe this as being able to consult a function $f(x)$ such that:

$$f(x) = \begin{cases} 1 \text{ , if } x \text{ is a solution;} \\ 0 \text{ , otherwise.} \end{cases} \tag{2.1}$$

Put differently, we have a function $f(x) : \{0,1\}^n \longrightarrow \{0,1\}$, and want to find x s. t. $f(x) = 1$, knowing there is exactly one such x. That is, we want to find the only input that evaluates to one. This is the full extent of our knowledge. Our trump is being able to evaluate the function f at any input.

Note that this access to the function $f(x)$ does not render the task of finding a solution trivial, as there are many questions for which it is easy to *verify* a solution, but not to *discover* one. This includes Sudoku, the knapsack problem, and other NP-complete problems.

However, it is easy to come up with a brute force strategy: if nothing else, we can find a solution by brute force, by evaluating inputs until we find our target. This is the best we can do using classical computers.

In this case, how many times do we expect to evaluate $f(x)$ before we find the solution? In the absence of any additional information, our best approach is to sample x uniformly at random. At each draw, we check whether $f(x) = 1$. If it is, we terminate. If not, we keep going. The probability of terminating exactly at the kth attempt is $1/N$ for all k, so the average number of attempts is given by the triangular number divided by the total number of items:

$$\sum_{k=1}^{N} \frac{1}{N} \cdot k = \frac{1}{N} \cdot \left(\frac{N(N+1)}{2} \right) = \frac{N+1}{2}. \tag{2.2}$$

The key point is that the computational cost (number of evaluations of f) is in:

$$\mathcal{O}(N). \tag{2.3}$$

In the worst case we check N items, which is also in $\mathcal{O}(N)$.

Now, onto the quantum version. We could simply reproduce the classical protocol, by replicating the classical algorithm - only instead of sampling from a distribution, we'd measure a quantum state. But while somewhat cooler, that would bring no advantage. Instead, we orchestrate convenient interference patterns, which allow us to increase the amplitude of success as fast as we increased the probability of success in the classical case (they both grow roughly by a constant with each extra oracle query). Because probabilities are squared amplitude norms, this means a quadratic speedup.

Indeed, a quantum algorithm can perform the searching task more efficiently, using a number of queries in $\mathcal{O}(\sqrt{N})$ - quadratically faster than the best performing classical one, as we have just seen.

Before proceeding, we generalize the searching problem by considering more than one marked item: instead of there being a single x s. t. $f(x) = 1$, there are M such x. We say they belong to a good subset X, and call these $x \in X$ the good items, or solutions, and all others bad, or wrong, items (x s.t. $f(x) = 0$, or equivalently $x \in X^c$, the complementary subset to X in $\{0, 1\}^n$).

In the previous case, we had $M = 1$. This gave rise to the classical query complexity of equation 2.3, which generalizes as

$$\mathcal{O}(N/M). \tag{2.4}$$

Now, in the quantum framework, the ability to identify a solution can be formulated as having access to an oracle operator $\hat{\mathbf{U}}_{\mathbf{f}}$ that identifies solution states by marking them somehow when it acts on them. For this reason, the terms solution states and marked states are often used interchangeably. As an example, this *marking* can mean assigning them (and only them) a π phase:

$$\hat{\mathbf{U}}_{\mathbf{f}} \mid x \rangle = (-1)^{f(x)} \mid x \rangle. \tag{2.5}$$

Note that this is a global, and hence immaterial, phase. However, it becomes consequential when the operator acts on a superposition of solutions and non-solutions. Also, up to a global phase, this is equivalent to reflecting about the solutions, i.e. reflecting the bad states; for searching purposes, we can do either (mark the good or the bad states), as long as we stay consistent and choose the number of amplification operations correctly.

Queries to the oracle in 2.5 are the quantum counterpart of evaluations of f; instead of saying whether $f(x)$ is 0 or 1, it acts on a state $\mid x \rangle$ encoding x and flips it i and only if $f(x) = 1$, i.e., it reflects the solutions and does nothing to other states. To physically realize this, we can use an auxiliary qubit initialized to $|\, - \,\rangle$ and act on it with an X gate controlled on f evaluated at the input x. This will kick a phase into the states when $f(x) = 1$: $|\, x \,\rangle \rightarrow -|\, x \,\rangle$.

Now, let's say we have a state that is a superposition of solutions and non-solutions,

$$|\, \psi_{\mathrm{A}} \,\rangle = \mathrm{A}\,|\, 0 \,\rangle^{\otimes n}, \tag{2.6}$$

where A is an initialization operator.

Naturally, we want to start at our best bet for a good solution, so A is our best guess operator for initializing the states. It should produce a superposition of possible solutions, with amplitudes reflecting our knowledge of their merits. Obviously, if we are aware that some state is not a solution, its amplitude should be zero. Conversely, if there's some guess we believe is very worthy, its amplitude should be large. A case where there are more specific insights is when classical heuristics are available that can also be applied to the quantum case [4]. Note that these insights are only worth using insofar as their associated state preparation cost does not override their utility.

But even if we don't have any a priori knowledge, as long as a solution exists, we can prepare an initial state that is a superposition of solutions and non-solutions: we do so by choosing A to be a Hadamard transform $H^{\otimes n}$, in which case $|\, \psi_{\mathrm{A}} \,\rangle$ is a uniform superposition of all states.

$$|\, \psi_{\mathrm{A}} \,\rangle = \mathrm{H}^{\otimes n}\,|\, 0 \,\rangle^{\otimes n} = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |\, x \,\rangle \tag{2.7}$$

Note that we assume that N is a power of 2 for simplicity, though a generalization is possible [4]. If the number of solutions is not a power of 2, we can use the Fourier transform instead, which can be generalized to non-power-of-two inputs [41]. The result is a uniform superposition of the states representing integers up to some arbitrary N; the elements differ only in their phases.

The point is, we can always write $|\, \psi_{\mathrm{A}} \,\rangle$ in terms of two projections: one into the good subspace, spanned by all solution states:

$$|\, \Psi_1 \,\rangle = \sqrt{\frac{1}{M}} \sum_{x \in X} |\, x \,\rangle, \tag{2.8}$$

and one into the bad subspace, spanned by all non-solution states:

$$|\, \Psi_0 \,\rangle = \sqrt{\frac{1}{N-M}} \sum_{x \in X^{\mathrm{c}}} |\, x \,\rangle. \tag{2.9}$$

Note that these subspaces are normalized. Thus, the wavefunction takes the form:

$$|\, \psi_{\mathrm{A}} \,\rangle = \sqrt{a}\,|\, \psi_1 \,\rangle + \sqrt{1-a}\,|\, \psi_0 \,\rangle. \tag{2.10}$$

The parameter a is what we call the *amplitude*. It depends on our state $|\, \psi_{\mathrm{A}} \,\rangle$. If $|\, \psi_{\mathrm{A}} \,\rangle$ is a uniform superposition, then a is the fraction of solutions to states,

$$a \equiv \frac{M}{N}. \tag{2.11}$$

In general, a is the initial probability of success: the probability of getting a good outcome x upon measuring $| \psi_A \rangle$.

Note it is $\sqrt{a}$, rather than a, what is typically called an amplitude in quantum mechanics; what we call *amplitude* here is actually a probability. Nevertheless, we preserve this designation, because it is widely used in the literature in the context of quantum amplitude estimation, with a few exceptions [1]. Note that the tasks of estimating these two quantities are equivalent when a is real and positive.

Often, it is convenient to rewrite equation 2.10 in terms of a single angular parameter: θ, which we call the Grover angle. It corresponds to the angle that $| \psi_A \rangle$ makes with $\left| \tilde{\psi}_0 \right\rangle$, or equivalently with the bad subspace:

$$\theta = \arcsin(\sqrt{a}) \leftrightarrow a = \sin^2(\theta). \tag{2.12}$$

The wavefunction then becomes:

$$| \psi_\mathrm{A} \rangle = \sin(\theta) \, | \psi_1 \rangle + \cos(\theta) \, | \psi_0 \rangle \, . \tag{2.13}$$

Now, suppose that we want to find a valid solution. We can attempt to do so by measuring this wavefunction, in which case we have a probability of success (here *success* is defined as finding a solution, or *good state*, upon measuring). This success rate is the same as that of the classical strategy in the same circumstances (no previous knowledge): it amounts to sampling an item at random in hopes of finding a solution. That this randomness is quantum doesn't bring any actual advantage.

Something different happens when we apply a special sequence of quantum operations to the state. These operations constitute what we call the Grover operator, defined as:

$$\hat{\mathbf{G}} = -\,\mathrm{A}\,\hat{\mathbf{U}}_\mathbf{0}\,\mathrm{A}^{-1}\,\hat{\mathbf{U}}_\mathbf{f}. \tag{2.14}$$

, where $\hat{\mathbf{U}}_\mathbf{0}$ assigns a π phase to the state $| \, 0 \, \rangle^{\otimes n}$, leaving all others unaltered. That is, it reflects the all-zero state, just as $\hat{\mathbf{U}}_\mathbf{f}$ reflected the pre-image of 1 under f. (Another common approach is to absorb the minus sign into $\hat{\mathbf{U}}_\mathbf{0}$, making it the operator that flips all states but zero.)

Because the A operator transforms that state into the uniform superposition $| \psi_\mathrm{A} \rangle$, and its inverse does the opposite, the sequence $\mathrm{A}\,\hat{\mathbf{U}}_\mathbf{0}\,\mathrm{A}^{-1}$ reflects the state $| \psi_\mathrm{A} \rangle$: it turns it into the all-zero state, then reflects this state, then transforms it back with an additional phase, but now with a sign attached.

$$A\hat{\mathbf{U}}_\mathbf{0}A^{-1} = \hat{\mathbf{U}}_\psi. \tag{2.15}$$

As such, the Grover operator can be seen as a sequence of two rotations.

Note that the implementation of $\hat{\mathbf{U}}_\mathbf{0}$ in terms of basic gates is straightforward: apply X gates to all qubits, then act with Z controlled on these qubits on an ancilla initialized to $| \, 1 \, \rangle$, then uncompute the X gates. The ancilla can then be ignored, having kicked back a π phase into the original register's all zero state. Similarly, the realization of the minus sign is conceptually simple: we can imprint a π phase into every computational basis state by performing similar controlled rotations, but with customized X gate patterns (applying it only to qubits which are $| \, 0 \, \rangle$ for that state).

For small enough amplitudes, the effect of this operator is to increase the amplitude; we call this *amplitude amplification*. It can be shown that m applications of Q result in the state:

$$\hat{\mathbf{G}}^m \mid \psi_{\mathrm{A}} \rangle = \sin((2m+1)\theta) \mid \psi_1 \rangle + \cos((2m+1)\theta) \mid \psi_0 \rangle . \tag{2.16}$$

Which means that our probability of success becomes:

$$|\langle \psi_1 \mid \hat{\mathbf{G}}^m \mid \psi \rangle|^2 = \sin^2((2m+1)\theta). \tag{2.17}$$

Crucially, our probability of finding a solution upon measuring the state has changed. In Grover's search algorithm, the advantage of being able to measure the wavefunction in 2.16 is that the probability of success is increased. As such, we'll have better odds of finding a solution, and will be able (on average) to do so more quickly, i.e. with less attempts.

This is called amplitude amplification, because we're amplifying the amplitude of some marked states. For small $\sqrt{a}$, after m iterations we'll basically have multiplied the probability of success a by $4m^2$ ($\sin(x) \approx x$ for x small so $\sin^2(c\theta) \approx c^2 a$ for small $c\theta$).

As long as the number of solutions M - and thus θ - is known, we can actually choose m to maximize the probability of success by bringing $(2m+1)\theta$ as close as possible to $\pi/2$, which would have that probability be unit. To achieve that, we would need:

$$(2m_{\text{ideal}}+1)\theta = \pi/2 \leftrightarrow m_{\text{ideal}} = \frac{\pi}{4\theta} - \frac{1}{2}. \tag{2.18}$$

Owing to the fact that our set up only allows integer m and θ is fixed by the problem, it isn't always possible to bring the angle to $\pi/2$ exactly. While some cases are deterministic (e.g. $M = 1, N = 4, \theta = \pi/6$), others are probabilistic (e.g. $M = 2, N = 4, \theta = \pi/4$). This means that more often than not, the operator of 2.14 is unable to achieve a 100% success rate, because it either under- or over-rotates (for the 2 best choices of m). Nevertheless, we can achieve a high enough one that by measuring a few times we can find a solution with near certainty; furthermore, the process is heralded, in the sense that we know when it succeeds.

Moreover, there exist strategies to de-randomize even the general case, yielding unit success probability [4]. One possibility is to use an ancilla to artificially reduce the probability of success on a continuous spectrum, transforming any particular case into one of the countable deterministic ones. It is easy to slightly decrease the probability of success a into some $\bar{a}$ by making the oracle enforce an extra condition: that an auxiliary qubit be in state $\mid 1 \rangle$. Since we can rotate this qubit however we want, this condition can change a on a continuum. We use that to change it into an $\bar{a}$ that results in an integer number of applications of Q: $\bar{m} = \lceil m_{\text{ideal}} \rceil$.

Another possibility is to customize the reflections in the Grover operation to add an arbitrary phase instead of a fixed one (π) as before. This is akin to performing non-integer applications of $\hat{\mathbf{G}}$. We can use the normal $\hat{\mathbf{G}}$ of equation 2.14, until we reach the iteration in which we would over-rotate. There, we adjust the reflection angle just enough that the amplified angle falls exactly in $\pi/2$ - whereas if we just applied the rigid Q, we would over-rotate.

Whatever approach we take, we are able to find a solution with an expected number of oracle queries in:

$$\Theta(\sqrt{1/a}), \tag{2.19}$$

or if using a Hadamard transform:

$$\Theta(\sqrt{N/M}). \tag{2.20}$$

Notice the *expected* keyword in the sentence above, just before equation 2.19. This is an average behavior, which already accounts for the non-null probability of failure when measuring $Q^m \mid \psi \rangle$, which happens when $m_{\text{ideal}} \notin \mathbb{N}$.

In short, we have presented a way to achieve a polynomial advantage in unstructured database search using quantum resources. Many problems of interest can be formulated in these terms. If additionally their best performing classical solution is either a brute-force algorithm or relies on heuristics that transfer to the quantum algorithm, the quantum approach provides a quadratic speed-up.

2.2 The original quantum amplitude estimation algorithm

We now switch focus to another objective. We want not to find a solution, but to *count* the number of solutions. This task is called *quantum counting*, and is a particular case of amplitude estimation. The latter is more general, as it can target any wavefunction of the form 2.10, with continuous a; that is, it is not restricted to an a of the form M/N for integer M, N. Clearly, N being known, we can convert estimates of a into estimates of M.

It can be shown that the Grover operator $\hat{\mathbf{G}}$ has 2 eigenvectors $\mid \Phi_{\pm} \rangle$, whose eigenvalues are:

$$\lambda_{\pm} = \exp(\pm i 2\theta). \tag{2.21}$$

Learning θ can be done by QPE on the operator $\hat{\mathbf{G}}$ defined in 2.14. Since we know the relation between θ and a, we can estimate the latter through the former. And naturally, we can estimate M as well.

For the QPE protocol, we need to implement controlled versions of $\hat{\mathbf{G}}$, and the (inverse) QFT$_K$. Here K is the order, which is defined by the number of qubits in the auxiliary register for the QFT: $K = 2^k$ for k qubits, which give k digits of precision at the output. The phase measurement made by the QPE circuit will collapse the wavefunction of the readout register into an encoding of the one of eigenphases, producing an outcome string that is determined by θ. We know how it is determined by θ, and so we can compute said θ using the outcome string. Once more, [4] generalizes the entire procedure to non powers of 2, including the QPE, but we consider the more common (and simpler) case.

In general, QPE produces:

$$\text{eigenvalue is } \exp(i2\pi\phi) \longrightarrow \text{QPE outputs } r = \phi \cdot K, \tag{2.22}$$

where again $K = 2^k$ with k the number of qubits we use for the auxiliary register (on which $\hat{\mathbf{K}}$ is controlled and the QFT† acts), and determines the precision of the estimate.

In our case $\phi = \theta/\pi$. Then, the measured QPE outcome r will be given by:

$$\begin{aligned}(2\pi\phi = \theta_1 \equiv 2\theta) \vee (2\pi\phi = \theta_2 \equiv -2\theta = 2\pi - 2\theta = 2(\pi - \theta)\\ \rightarrow \left(r = r_1 \equiv K\frac{\theta}{\pi}\right) \vee \left(r = r_2 \equiv K - K\frac{\theta}{\pi}\right)\end{aligned} \tag{2.23}$$

, where for QPE purposes we consider $\phi \in [0, 1[$ or equivalently $\theta_1, \theta_2 \in [0, 2\pi[$ since the output is an integer.

This allows us to compute:

$$a = \sin^2(\pi\frac{r}{K}) \tag{2.24}$$

$$M = aN \tag{2.25}$$

Note that the result is unchanged if we get r_2 from the minus sign eigenvalue, as $\sin(\theta) = \sin(\pi-\theta)$, which is really convenient (or else we wouldn't know how to interpret the data)! Also note that the minus sign on $\hat{\mathrm{G}}$ is indifferent for quantum searching, but in this case we'd get $N - M$ instead of M, which must be accounted for in the calculations.

The most important theoretical result of [4] states that estimating a in the manner described above uses $\mathcal{O}(K)$ oracle queries, and achieves an error upper bounded as:

$$|\bar{a} - a| \leq 2\pi s\frac{\sqrt{a(1-a)}}{K} + s^2\frac{\pi^2}{K^2} \tag{2.26}$$

with probability $8/\pi^2$ for $s = 1$ and $1 - 1/(2(s-1))$ for s any other positive integer.

The major takeaway is that the error in estimating a scales as:

$$\boxed{\epsilon \in \mathcal{O}(1/K)}, \tag{2.27}$$

where K is the number of oracle queries (equal to the Fourier order $K = 2^k$). We have also defined the error $\epsilon \equiv |\bar{a} - a|$.

Note that the scaling is given in terms of K (the quantum Fourier transform order), whereas in the previous section it was given in terms of a (the fraction of solutions). Both of them determine the number of queries used by the algorithm, which is the crucial resource.

Now, how does 2.27 compare to the best performing classical algorithm? Classically, we can estimate a by sampling from the original distribution and evaluating the function f for each sample. We then average these evaluations to get the expected value of $f(x)$, which is precisely our amplitude:

$$\frac{1}{N}\sum_x f(x) = \frac{1}{N}\sum_{x\in X} 1 = \frac{1}{N}M = a. \tag{2.28}$$

This is the typical sample mean framework. The error will be the standard error of the mean, which is equal to $\sigma/\sqrt{K}$, with σ the standard deviation of the original distribution - in this case, a binomial distribution - and K the number of samples/queries. The main point is that the error scales as:

$$\boxed{\epsilon \in O(1/\sqrt{K})}. \tag{2.29}$$

Meaning, we have a quadratic speedup for quantum amplitude estimation. When we use the quantum algorithm, the number of queries to the quantum oracle is quadratically smaller than the number of classical evaluations of f (or equivalently, than the number of classical samples we would need from the probability distribution, at which we must evaluate the function to compute the average), to achieve the same precision.

2.3 Outcome distribution and post-processing

In the previous section, we stated that the results of performing quantum amplitude estimation were probabilistic in most cases. Now, we would like to describe this probabilistic behaviour in full. Our interest in doing such stems from two main points. The first one is that it allows us to simulate the algorithm efficiently, without actually running the circuits (be it in classical or quantum simulators). This significantly lowers the cost of testing the algorithm. The second one is that thoroughly understanding how the algorithm works allows us to make the best of it.

We start by considering the outcome distribution of QPE. It estimates ϕ in $\exp(i2\pi\phi)$, but the actual measurement outcomes target $K\phi$, with K the order of the Fourier transform and the number of possible outcomes. If that is an integer, the outcome is deterministically $x = K\phi$, and we calculate ϕ as $\phi = x/K$. If it is not, the probability of each of the K outcomes, which we'll call x, increases with its accuracy, and observes the following expression:

$$P(\text{measuring } x \mid QPE(\phi)) = \frac{\sin^2(K\Delta\pi)}{K^2 \sin^2(\Delta\pi)}, \tag{2.30}$$

where Δ is a circular distance, and also the error in the estimate produced by x:

$$\Delta = \left|\phi - \frac{x}{K}\right| \mod 1. \tag{2.31}$$

This is an angular distance, divided by 2π.

In QAE, when we perform QPE, we measure one of two eigenvalues of the Grover operator, which changes the looks of the expressions above:

- One eigenvalue is $\exp(i2\theta)$, meaning that $2\pi\phi = 2\theta \leftrightarrow \phi = \theta/\pi$. In this case:
  - The exact case outcome is $x_0 = M\theta/\pi$.
  - We would calculate θ from an outcome x as $\pi x/M$.
  - Δ is defined as $\Delta = |\theta/\pi - x/M| \mod 1$, which measures the distance between θ/π (the phase-estimated angle divided by 2π) and x/M (the exact phase encodings representable by the QPE auxiliary register).

- The other eigenvalue is $\exp(-i2\theta)$, meaning $2\pi\phi = 2\pi - 2\theta \leftrightarrow \phi = 1 - \theta/\pi$. We're basicallly replacing $\theta \longrightarrow \pi - \theta$ as compared to the previous case:
  - The exact case outcome is $x_0' = M - M\theta/\pi = M(1-\theta/\pi) = M - x_0$ using the x_0 definition from the first case.
  - We would calculate θ from an outcome x as $\pi(M-x)/M = \pi - \pi x/M$.
  - Δ is defined as $\Delta = |1 - \theta/\pi - x/M| \mod 1$, which measures the distance between $1 - \theta/\pi$ and x/M.

    Alternatively we could say it measures the distance between θ/π and $1 - x/M$ (the calculated result, divided by π to get the QPE counterpart). These should be the same because they amount to calculating the distance between two angles $-\alpha$, $+\beta$ versus α, $-\beta$; the result is the same since we take the absolute value. Note that $1 - \alpha \mod 1 = -\alpha \mod 1$, which is easy to see by thinking of angles or using modular arithmetic (namely the distributive and identity properties):

$$\begin{aligned}
&(1+b) \mod 1 \\
&= ((1 \mod 1) + (b \mod 1)) \mod 1 \\
&= (0 + (b \mod 1)) \mod 1 \\
&= (b \mod 1) \mod 1 \\
&= b \mod 1
\end{aligned}$$

The second points in each of the two cases, which tell us how to calculate the unknown parameter θ from an outcome x, may seem problematic: in practice, we have no way of knowing which eigenphase was behind a measurement. How then can we know which formula to apply to calculate θ?

We cannot. However, as luck would have it, the object of our interest is not actually θ, but rather $a = \sin^2(\theta)$. Due to the symmetry of the sine function about $\pi/2$ in $[0, \pi]$, a can always be calculated as $a = \sin^2(\pi x/M)$, regardless of the underlying eigenvalue. Equivalently, it can be calculated as $a = \sin^2(\pi - \pi x/M)$.

Since we measure each eigenvalue with probability $1/2$, the probability of measuring some specific outcome x is given by the average:

$$P(\text{measuring } x \mid QAE(\theta)) = \frac{P(\text{measuring } x \mid QPE(\theta/\pi)) + P(\text{measuring } x \mid QPE(1-\theta/\pi))}{2}. \tag{2.32}$$

Or alternatively, as can be seen from formula 2.30 and the previous considerations on Δ:

$$P(\text{measuring } x \mid QAE(\theta)) = \frac{P(\text{measuring } x \mid QPE(\theta/\pi)) + P(\text{measuring } M-x \mid QPE(\theta/\pi))}{2}. \tag{2.33}$$

These equations give the exact probabilities of each possible QAE output. By sampling from a multinomial distribution defined by these probabilities, we can accurately reproduce the ideal behavior of the canonical quantum amplitude estimation algorithm under shot noise. This emulates the performance of a perfect quantum device while requiring a much shorter runtime, since it relies on a small number of simple analytical and sampling operations. Note that this not mean that this construction can replace the QAE algorithm, because it requires knowing the solution we're seeking: the calculations require specifying the amplitude parameter. In other words, these considerations are useful to study the behaviour of QAE, but not to solve the problem it tackles.

Now that we know how to mimic the workings of the QAE algorithm, let's do exactly that. Our results should look something like figure 1, which shows the outcome distribution of one execution of the algorithm.

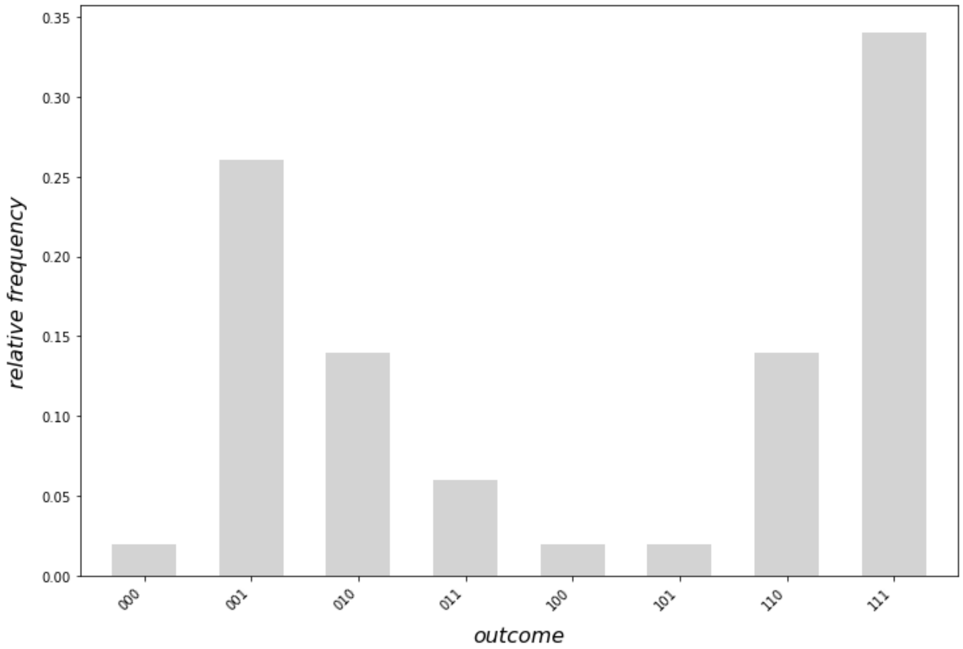


Figure 1: Bar plot of the outcomes of a single execution of the quantum amplitude estimation algorithm. The amplitude is $a = 0.3$, 50 measurements were used to compute the frequencies, and $K = 3$ auxiliary qubits were used for the quantum Fourier transform.

Now, how do we interpret these results? The first step is to translate them into the amplitude domain. Using the conversion presented earlier in this section, this is straightforward, and we get something like figure 2.

At this point, we want to pick a numerical estimator for the amplitude, which is our end goal. We can just pick the value corresponding to the highest relative frequency, which is approximately 0.1464 ($sin^2(\pi \cdot 1/8)$). But should we *really* do that?

Actually, 0.1464 is one of the few values that we are *absolutely sure* is wrong. That's because this amplitude could never have generated the plots in figures 1 and 2: it finds an exact representation in our k-qubit discretization. As such, it would have generated only 2 outcomes, both corresponding to the same

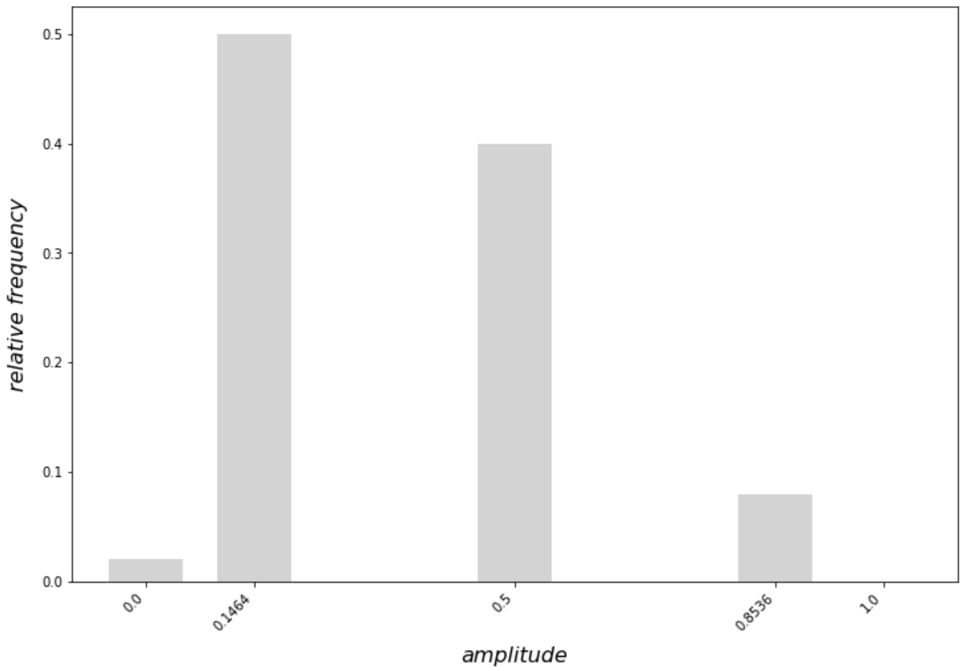


Figure 2: Bar plot of the amplitude estimates produced by a single execution of the quantum amplitude estimation algorithm. The amplitude is $a = 0.3$, 50 measurements were used to compute the frequencies, and $K = 3$ auxiliary qubits were used for the quantum Fourier transform.

amplitude - precisely 0.1464. The same goes for any other of the amplitude values in the x-axis of figure 2, for the very same reason they are present in said axis.

Out of those values, the mode is indeed our best choice. Clearly, it falls closer to the real value ($a = 0.3$) than any other. But we can actually do better than choosing on the grid, by availing ourselves of equation 2.32 (or 2.33), which describes how likely a given amplitude would be to generate an observed outcome. This allows us to consider amplitudes on a continuum: given any value, we can calculate how likely it would have been to produce the list of outcomes we observed. We do so by multiplying how likely it would have been to produce each of them.

We can then sweep over a much denser grid of amplitudes, without expending any extra quantum resources. All we need is the previously gathered data - in our example, that would be the experimental records from the fifty measurements of figure 1. The post-processing technique we are describing answers to the name of (maximum likelihood estimation (MLE)), and is suggested in [22] for this particular problem.

Even though no additional quantum resources are required, classical processing is. However, the optimization can be quite agile; more so than increasing the number of qubits to achieve the same precision. Even still, it becomes harder as the number of measurement repetitions increases. One easy trick that can aid it is to make a more localized search, by reducing the search range to the vicinity of the mode.

Since the most likely outcome should be the one with a smaller estimation error, the mode should be closer to the true value than any other grid point. Thus, it should suffice to search in a range centered at the mode, and limited to within the halfway distance between itself and its nearest ”grid points” (exactly representable amplitudes). This is also suggested in [22].

Since the results we actually obtain are susceptible to statistical noise, we may choose to be a bit more conservative and consider e.g. the full distance to each closest grid point. Figure 3 shows the results of doing this. Clearly, the estimate we obtain by maximum likelihood estimation is quite better than the one afforded by the grid.

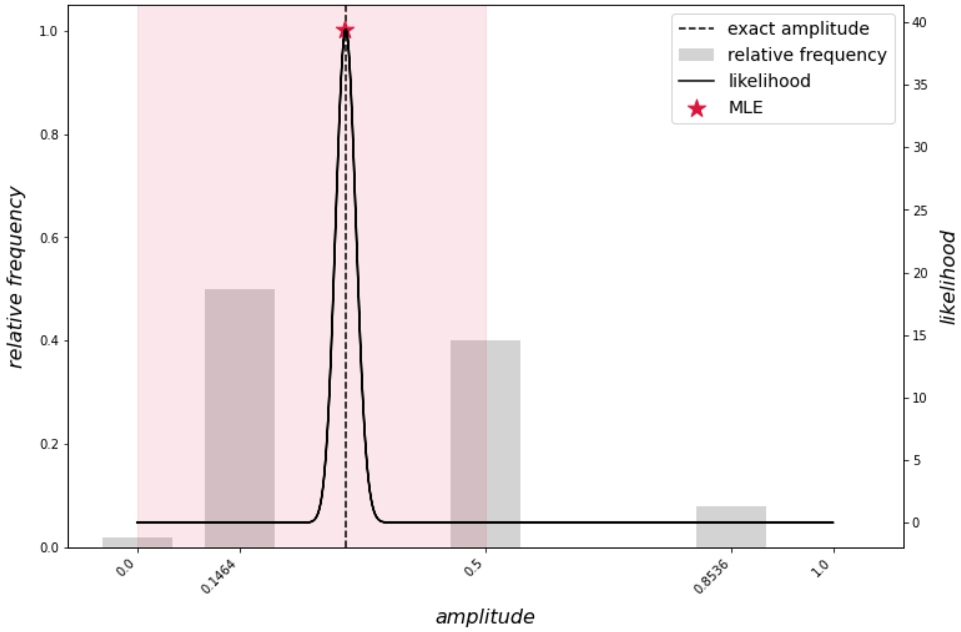


Figure 3: Plot of the likelihood function and maximum likelihood estimate produced by a single execution of the quantum amplitude estimation algorithm, juxtaposed with the original bar plot of the amplitude estimates produced by QPE. The optimization was performed by searching on the shadowed region, using brute force search on 100 grid points followed by Nelder-Mead maximization. The amplitude is $a = 0.3$, 50 measurements were used to compute the frequencies, and $K = 3$ auxiliary qubits were used for the quantum Fourier transform. The amplitude estimated by MLE was 0.298.

The likelihood landscape looks clearer if we look at the *logarithm* of the likelihood, usually called the *log-likelihood*. This is plotted in figure 4. We can see that due to the fact that the likelihood is zero, the log-likelihood plunges to $-\infty$ at the grid points (the graph is truncated at -400).

2.4 Toward noise-resilient amplitude estimation

Having presented the traditional quantum amplitude estimation algorithm, we would now like to turn our attention to alternatives that have been proposed over the past few years. All of them offer NISQ-friendlier substitutes for the canonical QAE protocol of [4].

We note that the names of the algorithms of the following sections are not very descriptive; most could be swapped without incurring inaccuracies, in the sense that all or almost all of them are *simplified*, *simpler*, *iterative*, and *faster*. Regardless, we preserve the original authors' epithets for ease of identification.

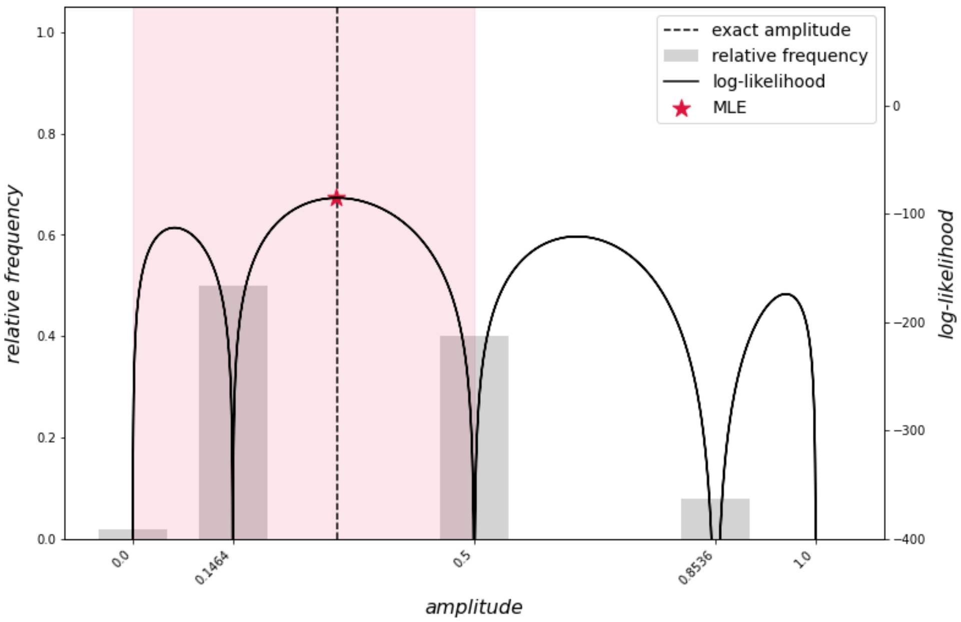


Figure 4: Plot of the log-likelihood function and maximum likelihood estimate produced by a single execution of the quantum amplitude estimation algorithm, juxtaposed with the original bar plot of the amplitude estimates produced by QPE. The optimization was performed by searching on the shadowed region, using brute force search on 100 grid points followed by Nelder-Mead maximization. The amplitude is $a = 0.3$, 50 measurements were used to compute the frequencies, and $K = 3$ auxiliary qubits were used for the quantum Fourier transform. The amplitude estimated by MLE was 0.298.

2.4.1 Maximum likelihood amplitude estimation

We start by the maximum likelihood (quantum) amplitude estimation (MLAE) algorithm of [56]. It is an approachable heuristic algorithm with a robust numerical performance, despite lacking rigorous theoretical support. A schematic representation of the algorithm can be seen in figure 5.

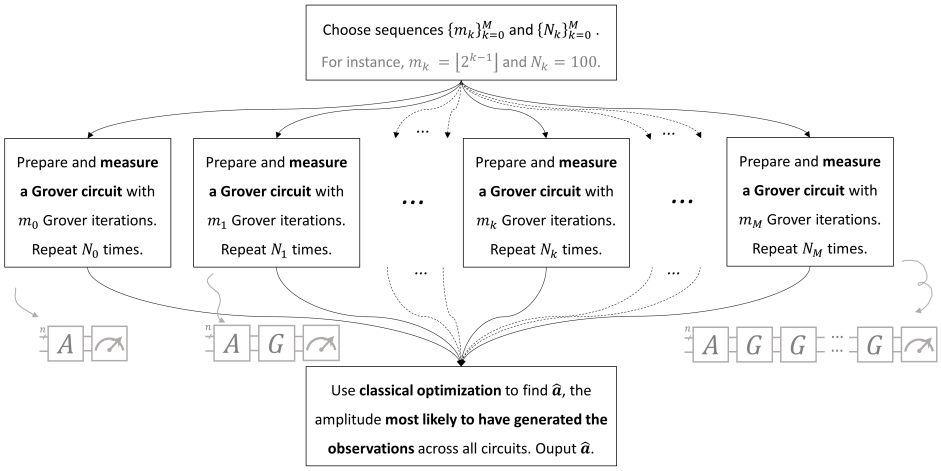

Choose sequences $\{m_k\}_{k=0}^{M}$ and $\{N_k\}_{k=0}^{M}$.
For instance, $m_k = \lfloor 2^{k-1} \rfloor$ and $N_k = 100$.
Prepare and **measure** **a Grover circuit** with m_0 Grover iterations. Repeat N_0 times.
Prepare and **measure** **a Grover circuit** with m_1 Grover iterations. Repeat N_1 times.
...
Prepare and **measure** **a Grover circuit** with m_k Grover iterations. Repeat N_k times.
...
Prepare and **measure** **a Grover circuit** with m_M Grover iterations. Repeat N_M times.
n
A
G
Use **classical optimization** to find $\hat{a}$, the amplitude **most likely to have generated the observations** across all circuits. Ouput $\hat{a}$.

Figure 5: Diagram summarizing the maximum likelihood quantum amplitude estimation algorithm of [56].

This algorithm is not to be confused with the original QAE algorithm, subjected to maximum likelihood post-processing. They both use maximum likelihood estimator, but differ strongly in how they collect the data to be used to compute the likelihood: in the latter they come from measurements on the conventional QAE circuit with the QFT, whereas here they come from measurements on multiple simpler circuits. Also note that maximum likelihood (quantum) amplitude estimation (MLAE) preceded the application of MLE to textbook QAE.

Indeed, MLAE forgoes QPE and controlled Grover operators entirely, and does not even estimate the Grover eigenphase directly. Instead, it attempts to directly infer the probability of success. i.e. the squared amplitude of the good projection.

Measuring Grover-style/QAA circuits amounts to sampling from functions of the Grover angle, this angle being straightforwardly connected to the parameter of interest (the amplitude). This means that we can use statistical inference to estimate these quantities. After initializing the state $A \mid 0 \rangle^{\otimes n}$ (notation of [Brass00]), the probability of success is a, which is precisely the amplitude to be estimated. This coincides with the only type of sampling we can perform classically.

Contrastingly, in the quantum case we can alter the wavefunction by applying Grover iterations. If we apply m Grover iterations to the initial state, the arguments of the sinusoidal functions in the likelihood will change, and with them the outcome distribution behind our measurement results. We expect that different m will bring varying information, much like in the experimental learning problems of my dissertation; and also that the freedom to control the likelihood function in this way will be beneficial for the inference process.

Clearly, $m = 0$ reduces to the classical case where we sample tentative solutions at random with a probability of succeding; the expected runtime for achieving error ϵ is in $O(1/\epsilon^2)$, as in classical Monte Carlo averaging. On the other hand, if m is the optimal number of applications of Q for amplitude amplification, the probability of measuring good states approaches 1. If we're to infer the probability of success from measurement data, we can use whatever combination of circuits we want, and consider the joint likelihood function they produce. The statistical model can easily be derived as a conjunction of independent events, whose individual probabilities follow from the definition of the Grover operator.

The approach of [56] is then to measure a chosen ensemble of circuits, and perform maximum likelihood estimation on the amalgam of measurement data. That is, we choose our estimator $\hat{a}$ to be the parameter value likeliest to have generated the outcomes we observed across all circuits. The authors consider running $M + 1$ different Grover circuits, which differ in the number of applications of the Grover operator. The kth circuit uses m_k applications of Q, and is measured N_k times.

The choice of the m_k and N_k hyperparameters is mostly empirical, though the authors present some lax theoretical arguments. They exclusively use a constant number of shots per circuit, ($N_k = N_{\text{shot}}$ for all k), and simple predetermined sequences $\{m_k\}_{k=0}^{M}$. It is shown numerically that even for these simple choices, the MLAE algorithm can surpass the standard quantum limit, signifying a quantum-enhanced estimation protocol. However, the Heisenberg limit is not reached.

One interesting and uncommon feature of this approach is its parallelizability. The measurement data can be collected in parallel across all circuits and shots, and must be joined together only for the post-processing step (which is a classical optimization routine). Other benefits of the algorithm are its flexibility and unassuming nature. For instance, the statistical model can be extended to consider noise or other circuit types. Indeed, the demands imposed on the structure of the circuits are little to none, at least in principle. This means they can be adapted in view of hardware constraints, such as coherence times. On the flip-side, the formal guarantees are scant; as such, there is no promise such adaptations won't erase the quantum advantage entirely.

2.4.2 Variational amplitude estimation

The variational (quantum) amplitude estimation (VAE) [47] algorithm was a spin-off stemming from MLAE: the essence of the strategy is preserved, but variational approximations are introduced to curb circuit depth. Note that even though the authors chose to introduce a variational strategy to MLAE in particular (and still more particularly, to MLAE with linearly incremental sequence (LIS)), it could just as well have been applied to any other algorithm relying on the same circuit type (as all others we have seen). For clarity, we will use variational maximum likelihood (quantum) amplitude estimation (V-MLAE) to signify MLAE-inspired VAE.

The idea is to periodically approximating sequences of Grover iterations as variational quantum circuits. This allows for setting a cap on how deep the circuits can become, whereas in the original algorithm depth increases indefinitely through the iterations.

However, the direct application of this strategy does not produce satisfactory results. While the performance shown in numerical tests is interesting, the variational cost is not considered at all; thus, the performance defects are exclusively due to limited expressibility. In practice, the variational approximation incurs significant costs, so this comparison is not fair.

As a matter of fact, the authors state - albeit without demonstrating it - that the cost of optimizing the variational approximation (tuning ansatz parameters) cancels out any advantages, rendering the performance even worse than that of classical Monte Carlo. To remedy this, they propose an *adaptive* variational algorithm, aimed at reducing the variational cost.

In this approach, prior to the optimization, the amplitude is re-scaled in such a way that the Grover circuit we want to implement corresponds roughly to the identity. Then, the variational parameters must be tuned to correct the error, rather than to approximate the circuit from scratch. This is a simpler optimization task.

For the approximate amplitude rescaling, an initial approximation of the amplitude is required. The authors propose using classical Monte Carlo to produce a rough estimate. The amplitude being what we want to discover, this may seem somewhat circular. However, it is a valid strategy, as long the associated cost is considered in the performance analysis - which it is.

Naturally, if the strategy relied entirely on the classical Monte Carlo estimate, it could never show quantum-enhanced performance; thus, as long as properly accounted for, such an estimate could never bring an unfair advantage. At worst, it could negatively affect the performance of the algorithm.

In this case, it does not - on the contrary, it benefits the performance. The introduction of the adaptive preparation step leads to a additive overhead in the complexity, in spite of which the algorithm exhibits a competitive performance (for moderately large N_q). Importantly, this competitiveness endures even after the variational cost is considered.

The contribution of this paper is then to provide a quantum speed-up roughly as large as that of [56], while enforcing an upper limit on circuit depth. Given the limited coherence times of physical quantum devices, this is an interesting attribute. It comes at the expense of an additive overhead, which is due to a classical preparatory step.

2.4.3 Low depth quantum algorithm estimation (Power law/QoPrime)

The power law (quantum) amplitude estimation (PLAE) algorithm is yet another approach based on MLAE, put forward in [19]. It likewise relies on measurement data from multiple circuits, differing only in how these circuit are constructed: the original (linear or exponential) Grover power schedules of [56] are replaced with a power law sequence, $m_k = k^{f(\beta)}$. The algorithm is proven correct under certain regularity assumptions.

Here $\beta \in]0, 1]$ is a controllable parameter negotiating a balance between the overall cost N and the maximal circuit depth D, both quantified in terms of oracle queries. More specifically, these quantities evolve according to $N \in O(1/\epsilon^{1+\beta})$ and $D \in O(1/\epsilon^{1-\beta})$ respectively, where ϵ is an additive error. Note that the product ND is a constant.

For $\beta = 1$, we recover the classical algorithm, whereas $\beta \longrightarrow 0$ comes closest to the best performing schedule of [56], as well as the ideal quantum-enhanced performance. When β takes in-between values, we are working in an intermediate regime where part of the quantum advantage is traded for hardware-friendliness. Volunteering for this exchange allows one to controlledly concede part of the quantum advantage; this is in the practitioner's best interest, as often holding on to its entirety would be counterproductive. In the absence of fault-tolerant quantum devices, failing to compromise according to hardware limitations is likely to result in unchecked erasure of the quantum advantage. By means of circuit tailoring, one may extract the maximum quantum advantage that a faulty quantum device can offer, by working just until where its coherence allows.

In the same paper, the authors propose a second algorithm, dubbed QoPrime (quantum) amplitude estimation (QPAE). It accomplishes a behaviour identical to that of PLAE, but by different means. While it also relies on an ensemble of relatively shallow circuits, it chooses and processes them differently, leveraging a theorem borrowed from number theory. In terms of formal guarantees, there are two main differences between PLAE and QPAE. One is that the latter restricts the values of β to a discrete set; and the other is that it relinquishes the assumptions that PLAE necessitates.

In practice, PLAE tends to show good performance - indeed, numerical tests had it outperform QPAE. However, its formal guarantees may seem unsatisfactory, due to relying on premises that may not hold or be hard to ascertain. This makes the generality of QPAE theoretically interesting: it proves that the cost-depth trade-off presented earlier can be achieved while assuring correctness.

Both PLAE and QPAE do well in noisy simulations, demonstrating a competitive performance against other state of the art algorithms (which, unlike them, are not equipped to deal with decoherence).

2.4.4 Amplitude estimation, simplified

"Quantum amplitude estimation, simplified" (to be abbreviated as AES), an algorithm presented in [1], is a rigorous take on QFT-free QAE, with perfect asymptotic guarantees but a daunting constant factor. Figure 5 shows a schematic representation of the algorithm.

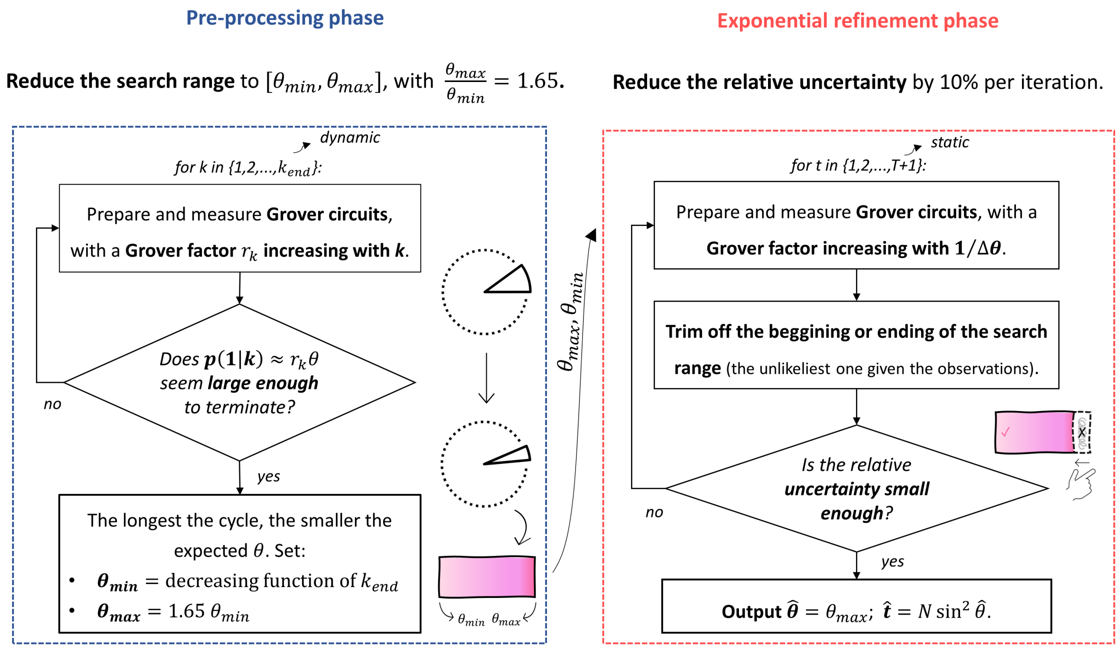

Pre-processing phase
Exponential refinement phase
Reduce the search range to $[\theta_{min}, \theta_{max}]$, with $\frac{\theta_{max}}{\theta_{min}} = 1.65$.
Reduce the relative uncertainty by 10% per iteration.
dynamic
for k in {1,2,...,k_{end}}:
Prepare and measure **Grover circuits**, with a **Grover factor** r_k **increasing with** $\boldsymbol{k}$.
Does $\mathbf{p(1|k)} \approx r_k\theta$ seem **large enough** to terminate?
no
yes
The longest the cycle, the smaller the expected θ. Set:
$\boldsymbol{\theta_{min}}$ = decreasing function of k_{end}
$\boldsymbol{\theta_{max}} = 1.65\, \theta_{min}$
θ_{min} θ_{max}
$\theta_{max}, \theta_{min}$
static
for t in {1,2,...,T+1}:
Prepare and measure **Grover circuits**, with a **Grover factor increasing with** $\mathbf{1/\Delta\theta}$.
Trim off the beggining or ending of the search range (the unlikeliest one given the observations).
Is the relative **uncertainty small enough**?
no
yes
Output $\hat{\boldsymbol{\theta}} = \theta_{max}$; $\hat{\boldsymbol{t}} = N \sin^2 \hat{\theta}$.

Figure 6: Diagram summarizing the quantum counting algorithm of [1] (the generalization to quantum amplitude estimation is straightforward). The search range width, defined as $\Delta\theta = \theta_{\text{max}} - \theta_{\text{min}}$, is rescaled as $\Delta\theta/$min to produce the relative uncertainty. The real value is contained within the determined confidence region up to a bounded probability of failure, which can be controlled by the number of repetitions for the Grover circuit measurements. The correctness of the pre-processing phase relies on proper choices of the Grover factors r_k and the termination criterion, which should jointly guarantee that with high probability the termination is timely. To be timely is to select a k_{end} for which $\theta_{\text{min}}(k_{\text{end}}) < \theta < \theta_{\text{max}}(k_{\text{end}})$.

Despite being originally called "approximate quantum counting, simplified", it can be straightforwardly applied to amplitude estimation. As a matter of fact, this is done in the paper. As such, we adapt its name accordingly, so that it better matches others and the focus of this work. We will proceed similarly with respect to other algorithms discussed herein.

This is an interesting proof-of-concept algorithm, because it was the first to rigorously prove that QFT-free QAE could achieve the optimal scaling (full quantum advantage) of [4]. Other papers from the same year had attempted such strategies, but did not offer as strong formal guarantees, instead focusing in simpler theoretical arguments and numerical analyses). Since QPE is usually associated with Shor-type exponential speed ups, it's theoretically nice to dissociate it from the Grover-type quadratic advantage of QAE.

The obtained scaling is, as in [4], $\mathcal{O}(\epsilon^{-1}\sqrt{N/M})$. They also include a controllable probability of failure δ, giving for the complete query complexity:

$$N_q \in \mathcal{O}(\epsilon^{-1}\sqrt{N/M}\log(\delta^{-1})), \tag{2.34}$$

where ϵ is a relative error:

$$M(1-\epsilon) < \hat{M} < M(1+\epsilon). \tag{2.35}$$

For general amplitude estimation, replace $M/N \longrightarrow a$, or actually $\sqrt{M/N} \longrightarrow a$ in the language of this paper. (This paper actually means the amplitude in amplitude estimation literally, which is interesting because most don't and estimate the probability instead. It is still assumed that the amplitude is a real number, but they note that there's no loss of generality because the phases can be absorbed into the basis states.)

The downside is that although the Heisenberg limit $\epsilon \in \mathcal{O}(N_q^{-1})$ is attained, the involved constant factor is unfavorable [Naka20, Grink21]. That is, k is large in $\mathbb{E}(\epsilon) = kN_q^{-1}$. The algorithm design seems to mostly be concerned with satisfying asymptotic scaling requirements, which makes it somewhat unsuitable for practical use.

Since this algorithm is quite elaborate, we'll start by making a shallow description of it. It seeks to learn the Grover angle θ up to some relative error. Before proceeding, note that this angle can be made arbitrarily small by adding some dummy unmarked items to the database, or more generally by a single qubit rotation and a logical redefinition of markedness. For instance, to reduce the amplitude to at most 0.1 we can rotate an ancilla qubit so that the amplitude of state $|\ 1\ \rangle$ is 0.1. We then define a marked state as a conjunction of the previous condition with the ancilla being in state $|\ 1\ \rangle$.

Small angles has some nice properties. First, the Grover probability $\sin^2(r\theta)$ will be more fine grained, reducing the impact of the rounding errors that result from only being capable of performing an integer number of iterations. Second, we can use $\sin(\theta) \approx \theta$ to simplify analysis. Finally, it may allow us to work in a regime where multiples of the angle tend to behave predictably and intuitively, evolving by small increments that monotonically change the sine and cosine.

It is not these properties per se that offer an advantage; indeed, merely scaling the amplitude down would if anything make it harder to resolve (using standard sampling). Rather, the convenience comes from how this structure can facilitate the orchestration of a procedure benefitting from it.

The (quantum) amplitude estimation, simplified (AES) is one such procedure, and it consists of 2 main phases: a pre-processing stage, and an exponential refinement stage. Both of them rely on Grover circuits with varying numbers of Grover iterations and shots. By Grover iterations we mean sequential applications of the Grover operator $\hat{\mathbf{G}}$. We leave the unguarded term iterations for the iterations of the AES algorithm, which as we shall see rely on these circuits.

Such circuits produce a parametrized wavefunction depending on the unknown $\theta = \arcsin(a)$ (following the usual convention rather than this papers' $\theta = \arcsin(\sqrt{a})$):

$$G^{(r-1)/2} \mid \psi \rangle = \sin(r\theta) \mid \psi_1 \rangle + \cos(r\theta) \mid \psi_0 \rangle \,. \tag{2.36}$$

Apart from the dependence on the target parameter, we have a dependence on the number of Grover iterations, here written in terms of r, which we will hereafter call the Grover factor. It is the factor that multiplies the angle in the sinusoid's arguments. Clearly, this is only physically viable for r an odd integer, since the operator can only be applied an integer number of times. Note that the Grover factor is proportional to the number of Grover iterations, so they are interchangeable when considering the asymptotic scaling of any quantity with respect to the number of queries. More commonly, the Grover factor is written in terms of the number of Grover iterations m as $r = 2m + 1$.

We will now attempt a more in depth explanation of how the algorithm works.

The goal of the pre-processing stage is to establish a restricted search range $[\theta_{\text{min}}, \theta_{\text{max}}]$ that contains θ with a bounded and controllable probabiltity of failure. The restriction is that the minimum and maximum angles differ by a fixed multiplicative factor, i.e. $\theta_{\text{max}}/\theta_{\text{min}}$ is a predetermined constant. This is achieved iteratively. Each iteration k relies on measuring an ensemble of identical Grover circuits, the Grover factor r_k being determined by the iteration number and increasing exponentially with it. When working with small angles, this means that the probability p_{marked} of observing a marked state increases from one iteration to the next as $\sin^2(r_k\theta) \approx (r_k\theta)^2$ for growing r_k.

The key thing to note here is that the argument is proportional to the Grover angle θ itself. Consequently, the value of θ dictates how fast p_{marked} increases as the iterations progress. We then continuously perform iterations until this probability crosses some threshold. Based on how many iterations were necessary to achieve that, we can identify the region where θ should reside. The longer it takes to terminate, the smaller the angle θ probably is (again for small angles). With this in mind, we define $\theta_{\text{min}} \equiv \theta_{\text{min}}(k_{\text{end}})$ and $\theta_{\text{max}} \equiv \theta_{\text{max}}(k_{\text{end}})$ as decreasing functions of the number of iterations before termination, k_{end}. In our definition, we make them respect the intended ratio $\theta_{\text{max}}/\theta_{\text{min}}$. Correctness ($\theta_{\text{min}}(k_{\text{end})} \leq \theta \leq \theta_{\text{max}}(k_{\text{end}})$) can be safeguarded by appropriately choosing the pace at which the Grover factor is increased through the iterations, and the termination criterion.

Notably, the impact of the latter will be affected by the fact that we can only estimate p_{marked} up to some statistical noise, by sampling from a binomial distribution with parameter p_{marked}. As a result, only probabilistic guarantees can be made, as there is a risk that we will terminate too soon or too late independently of the underlying probabilities. However, we can make this risk arbitrarily small by measuring the Grover circuit for a high enough number of shots at each iteration, since then the relative frequency $\hat{p}_{\text{marked}}$ should approach the asymptotic frequentist value p_{marked}. This is helped by the fact that there is a range of correct termination times between the first one that to be large enough one and the first one to be too small (as long as the exponential increasing rule for k is chosen adequately, i.e. the base is small enough) .

Upon exiting this cycle, we have narrowed down the region we expect to contain θ, assuring its lower and upper extremes θ_{min}, θ_{max} are roughly of the same order of magnitude. The objective of the second phase is to shrink the uncertainty further, but now leveraging these facts. Just like with the previous loop, these refinements are performed iteratively and rely on Grover measurements; only this time, the Grover factor is not chosen according to the iteration number. Rather, it is determined by the values of θ_{min} and θ_{max}.

In the second phase, the relative uncertainty (the scaled width of the interval) is reduced by a constant factor at each iteration (independent of the iteration number). More specifically, we pick one of the two boundaries to move inward by a small amount according to a fixed rule. The task entrusted to Grover measurements is to answer the question: which of them should we move? Obviously, we want to trim one far end of the interval that does not contain θ. For that, we need to perform experiments that induce a fairly different expected behaviour depending on whether the real θ is located on the lower margin or on the higher margin of our starting search range.

For cases where the real parameter isn't close to either extremity, both choices are safe, since we're only pruning the corners. While this may sound less than optimal at first glance, opting for an aggressive shrinking rule would be unwise: this safety gap serves as a statistical insulant between the extreme probability cases. For instance, if we wanted to cut down the interval by half, it would be extremely hard to discern between the values immediately to the right of the center and those immediately to its left. Assigning a binomial parameter to one of two intervals to a fair degree of confidence is very hard if the separation distance between those intervals is scant. Therefore, our greediness would have us making an exaggerated amount of trials.

With this in mind, we can discuss the design of the Grover factor. We want to tell which is case more likely: $\theta \approx \theta_{\text{min}}$, or $\theta \approx \theta_{\text{max}}$, after which we'll contract the unlikely margin. Ideally, the strict equality cases would result in as different as possible Grover probabilities, which are determined by the Grover factor. In other words, we would like $\sin^2(r\theta)$ to near unit for one of these cases, and zero for the other. Equivalently, we want to choose an r such that the multiple $r\theta$ is around $l\pi/2$ for one, and around $l\pi$ for the other (for $l \in \mathbb{Z}$).

The hard part is creating and analysing a rule that will roughly achieve that, i.e. a function that given θ_{min}, θ_{max} produces an r that makes them behave as distinctively as possible. It is at this point that we need the interval to be fairly narrow, meaning that the "paces" of the angles in the interval aren't too mismatched when incrementing them around the unit circle. This makes it possible to establish the a good number of iterations. Its optimality suffers from 3 problems: some rounding errors due to the restrictions in choosing r, statistical noise, and the fact that θ won't actually sit at the exact edge of the frontier. Still, the correct choice can be identified with high probability while satisfying the scaling requirements. Once more, the failure probability can be controlled by means of the number of measurement repetitions.

This concludes the considerations on the procedure to efficiently reduce the search range exponentially fast, which can then be repeated as needed. The second and final phase comes to a close when we are satisfied with the width of the interval, i.e. when the probable range is as small as the accuracy we want to achieve.

2.4.5 Simpler amplitude estimation

Simpler amplitude estimation comes from [61]. It is a straightforward algorithm supported by indirect theoretical arguments and a few isolated numerical evaluations. Unlike others, it uses Hadamard tests (cf. appendix A) instead of using the Grover operator directly. A diagram of the algorithm can be found in figure 7.

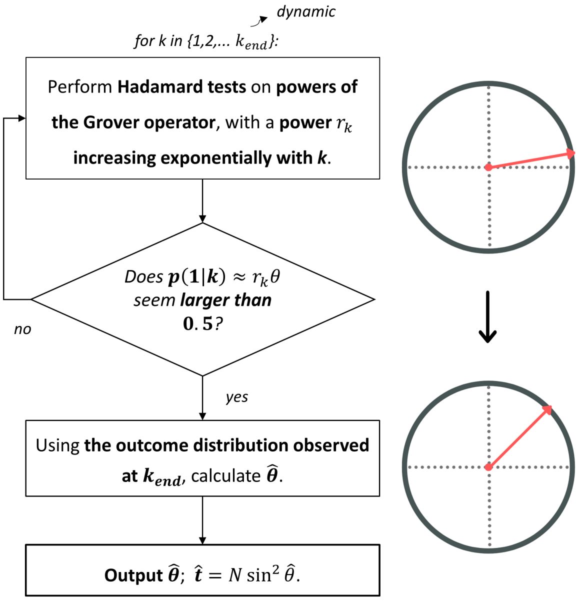

dynamic
for k in {1,2,... k_{end}}:
Perform Hadamard tests on powers of the Grover operator, with a power r_k increasing exponentially with k.
Does $p(1|k) \approx r_k\theta$ seem larger than 0.5?
no
yes
Using the outcome distribution observed at k_{end}, calculate $\hat{\theta}$.
Output $\hat{\theta}$; $\hat{t} = N\sin^2\hat{\theta}$.

Figure 7: Diagram summarizing the quantum counting algorithm of [61] (the generalization to quantum amplitude estimation is straightforward).

This strategy is based on successive "Hadamard tests", which require controlled Grover iterations and a single readout qubit (the control). The circuits are executed iteratively with exponentially increasing powers of the Grover operator, until the probability of finding the readout qubit in state 1 roughly crosses 50%. Analysing the circuits, analytical formulas can be obtained for the outcome probabilities and functions thereof. They depend on the Grover exponent and - crucially - on the Grover angle θ. As such, inverting them at the termination step gives an estimate for θ.

On the one hand, they prove that termination occurs after $O(\sqrt{t/N})$ queries, under the assumption that probabilities can be calculated exactly. This is the same complexity as the canonical algorithm of [Brass00] with respect to t and N (though usually we are more interested in how the queries scale with the estimation error, rather than the database parameters). On the other hand, they show numerically that in many cases the results are more accurate than those of the canonical algorithm - namely for small amplitudes, which the latter struggles to resolve. Based on these two facts, they claim their algorithm is more reliable and more practical than the original one.

As compared to textbook QAE, this algorithm does present a significant reduction in hardware requirements for the considered cases. However, the theoretical analysis is limited, and the numerical experiments are not very extensive. In particular, the theoretical results concern solely the idealized runtime of the algorithm, disregarding what is achieved in that runtime as well as the existence of shot noise; the scaling the canonical QAE simulations are performed only for very few qubits; the behaviour of the canonical algorithm seems to display anomalous behaviour, possibly due to the limited number of shots and use of single runs; and the asymptitoc behaviour of the error with respect to the number of queries, typically the most relevant performance metric in metrology, is not considered at all (be it analytically or numerically).

Furthermore, simpler (quantum) amplitude estimation (SAE) requires controlled operations, unlike most other near-term protocols. The author does not discuss the advantages of using Hadamard tests or the suggested processing formulas, as compared to simpler alternatives (elementary Grover circuits and a direct inversion of the first obtained formula for the sucess probability).

2.4.6 Iterative amplitude estimation

We have now gotten to the iterative algorithm of [22]: a greedy algorithm with a solid theoretical analysis, whose error scales almost as favorably as in the fault-tolerant algorithm (differing from the latter only by a double-logarithmic factor). Figure 8 schematic representation of the algorithm can be seen in figure 5.

The iterative quantum amplitude estimation algorithm relies on sequential refinements of a confidence interval for the Grover angle θ. These refinements are based on adaptive measurements, which are carried out successively until the uncertainty in the amplitude is as small as desired.

The measurement operations are performed on simple Grover circuits (i.e. pure amplitude amplification, without controlled gates), and governed by the usual experimental control: the number of applications of the Grover operator. This number is chosen with two goals in mind: to be as informative as possible, and to ensure that the measurement outcomes allow for unequivocally locating the true parameter (up to shot noise).

Intuitively, we want to learn about an angle by sampling from a sinusoidal function of it. We can amplify this angle before sampling; that is, multiply the argument of the sinusoid by a factor. Naively, we may be tempted to make this factor as large as possible, to increase the sensitivity of the outcome distribution with respect to the angle. However, because the sinusoidal function is cyclical, this will erase lower order information, as a measurement does not tell us how many cycles have been completed. Thus, establishing the amplification schedule requires a compromise.

More technically, what we want is to maximize the information (in this case, the Fisher information) while allowing for unambiguously inverting the likelihood function in the relevant domain. The Grover factor r shows up in the likelihood function as $P(1 \mid \theta) = \sin^2(r\theta)$. Increasing r amplifies the remainder

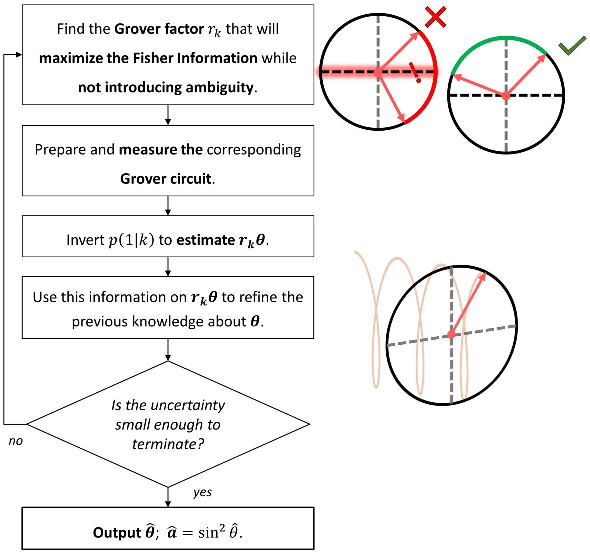

Find the **Grover factor** r_k that will **maximize the Fisher Information** while **not introducing ambiguity**.
Prepare and **measure the** corresponding **Grover circuit**.
Invert $p(1|k)$ to **estimate** $\boldsymbol{r_k\theta}$.
Use this information on $\boldsymbol{r_k\theta}$ to refine the previous knowledge about $\boldsymbol{\theta}$.
Is the uncertainty small enough to terminate?
no
yes
Output $\hat{\boldsymbol{\theta}}$; $\hat{\boldsymbol{a}} = \sin^2\hat{\theta}$.

Figure 8: Diagram summarizing the iterative quantum amplitude estimation algorithm of [22], which works by sequentially refining a confidence interval on the Grover angle θ. At each iteration k, the information on the amplified angle $r_k\theta$ makes it easier to resolve the least significant digits of θ, while the previously gathered knowledge on θ robustly identifies the most significant ones. In the setting of IAE, the utility of each measurement depends only on the Grover factor r_k. Thus, it can be maximized with respect to r_k at each iteration, with the caveat that for inference purposes the entire interval must lie either in the upper half-circle or in the lower one (so that the likelihood function is invertible and the data are not ambiguous).

$P(1 \mid \theta) \mod 2\pi$, which makes it easier to resolve. By choosing this factor carefully, we can locate the whereabouts of $r\theta \mod 2\pi$ in the full interval $[0, 2\pi]$, up to a desired precision and failure probability (which are due to sampling noise).

Clearly, this is not sufficient for locating θ in the same interval. However, this insight can be joined together with the coarser information from previous iterations, starting with the always-valid $\theta \in [0, \pi/2]$. The smallest and largest allowed values for θ that can result from combinations of the "integer" (multiple of 2π) and "fractional" ($\mod 2\pi$) parts serve as refined limits for the interval. These refinements can take place successively until the interval is as small as the target accuracy.

The resulting algorithm's scaling is described by the following upper bound:

$$N_q < \frac{50}{\epsilon} \log\left(\frac{2}{\delta} \log_2\left(\frac{\pi}{4\epsilon}\right)\right). \tag{2.37}$$

This has the intended behaviour $N_q = O(1/\epsilon)$, apart from the double-logarithmic multiplying factor $\log(2/\delta \cdot \log_2(\pi/(4\epsilon)))$.

Note that the above inequality is achieved using the Hoeffding's inequality. The use of Clopper-Pearson's intervals cuts the 50 constant down to 14, at the expense of more complicated calculations for the confidence intervals at each iteration; while a numerical analysis reduces each of these constants by a factor of roughly 10, or 50 if considering the average performance rather than the worst case.

2.4.7 Modified iterative amplitude estimation

The iterative quantum amplitude estimation algorithm from the previous section sparked some interest in possible variants or enhancements. One of them was put forth by [17]: modified (quantum) iterative amplitude estimation (M-IAE).

In both versions, the schedule of failure probabilities across rounds assures a user-specified minimum probability of success; the difference between them lies on *how* they follow through on this guarantee. The original proposal assures the probability of failure is upper bounded by the *same* constant at each round, so the rounds are equally responsible for the overall probability.

In contrast, the modified algorithm distributes the probability of failure biasedly, being stricter with earlier rounds, whose circuits are shorter. This is a more favorable arrangement, which brings an improved scaling.

2.4.8 Faster amplitude estimation

The faster (quantum) amplitude estimation (FAE) algorithm was introduced in [42]. It is a two-stage algorithm, with an asymptotic performance similar to the best to date but a smaller multiplying constant. Refer to figure 5 for a diagrammatic view of the protocol.

2.4.9 Summary

Table 1 summarizes of the algorithms presented in this section.

2.5 Application to Monte Carlo integration

We now present a summary of the basics of quantum-enhanced Monte Carlo integration, which is based on QAE. This section will follow [56].

The goal is to estimate the expectation value of a function $f(x)$ given a probability distribution $p(x)$ by averaging over samples. Asympmtotically, this amounts to integrating $f(x)$ with respect to $p(x)$:

$$\mathbb{E}_{p(x)}[f(x)] = \int_{\Omega} f(x)p(x)\mathrm{d}x, \tag{2.38}$$

where $\Omega \subset \mathbb{R}^d$ is the integration interval, and $x \in R^d$. A Monte Carlo estimate of this integral can be obtained by importance sampling $\{x_i\}_{i=0}^{N-1} \sim \pi(\cdot)$ for some distribution of choice $\pi(x)$ and averaging:

$$\mathbb{E}_{p(x)}[f(x)] \approx \frac{1}{N}\sum_{i=0}^{N-1} \frac{f(x_i)\cdot p(x_i)}{\pi(x_i)}. \tag{2.39}$$

Ideally, the importance distribution $\pi(x)$ would resemble $p(x)$, but in practice sampling from the latter may not be feasible. Expression 2.39 still stands, despite being less efficient with respect to the

	Key idea	Parallelizable	Circuits	Main strength(s)	Main weak-ness(es)	Complexity
QAE [4]	QPE on Grover operator	no	QPE	optimal complexity	very deep circuit; noise oblivious	$N_q \in \mathcal{O}(\frac{1}{\epsilon} \log \frac{1}{\alpha} \cdot a^{-1})$
MLAE [56]	heuristic measurements ⟶ statistical estimation	fully	QAA	simplicity; solid numerical performance	no formal guarantees; unchecked circuit growth	best case | observed: $N_q^{(\text{LIS})} \sim \epsilon^{-0.75} \mid \epsilon^{-0.76}$ $N_q^{(\text{EIS})} \sim \epsilon^{-1} \mid \epsilon^{-0.88}$
V-MLAE [47]	MLAE with variational approximation	fully	QAA	solid numerical performance; limits circuit depth	no formal guarantees; cost overhead	similar to MLAE except for EIS-observed (untested)
AES [1]	rough localization ⟶ exponential refinement	partly	QAA	optimal complexity	large cost offset; noise oblivious	$N_q \in \mathcal{O}(\frac{1}{\epsilon} \log \frac{1}{\alpha} \cdot a^{-1})$
SAE [61]	amplify ⟶ invert probability	partly	Hadamard tests	–	inconclusive demonstrations	$N_q \in \mathcal{O}(a^{-1})$
IAE [22]	watchful choice of Fisher information	partly	QAA	nearly optimal complexity	unchecked circuit growth; noise oblivious	$N_q \in \mathcal{O}(\frac{1}{\epsilon} \log(\frac{1}{\alpha} \log \frac{1}{\epsilon}))$
M-IAE [17]	IAE but distribute shots more favorably	partly	QAA	optimal complexity	same as IAE	$N_q \in \mathcal{O}(\frac{1}{\epsilon} \log \frac{1}{\alpha})$
FAE [42]	complementary measurements ⟶ invert probability	partly	QAA	nearly optimal complexity	unchecked circuit growth; noise oblivious	$N_q \in \mathcal{O}(\frac{1}{\epsilon} \log(\frac{1}{\alpha} \log \frac{1}{\epsilon})$

Table 1: Table summarizing the characteristics of a selection of QAE algorithms. All circuits are based on Grover operations, but differ in how they are structured. Complexities were demonstrated analytically unless otherwise stated, and denote the number of queries N_q necessary to estimate an amplitude a to error at most ϵ with probability at least $1 - \alpha$.

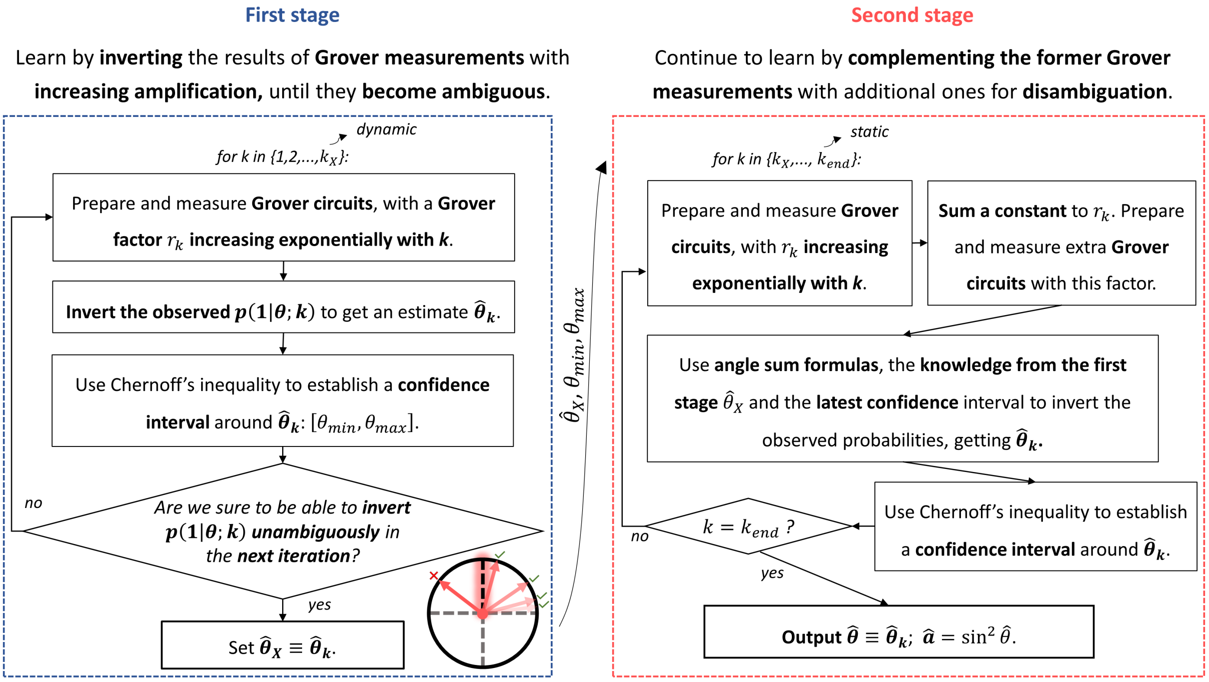

First stage
Learn by inverting the results of Grover measurements with increasing amplification, until they become ambiguous.
dynamic
for k in {1,2,...,k_X}:
Prepare and measure Grover circuits, with a Grover factor r_k increasing exponentially with k.
Invert the observed $p(1|\theta;k)$ to get an estimate $\hat{\theta}_k$.
Use Chernoff's inequality to establish a confidence interval around $\hat{\theta}_k$: $[\theta_{min}, \theta_{max}]$.
no
Are we sure to be able to invert $p(1|\theta;k)$ unambiguously in the next iteration?
yes
Set $\hat{\theta}_X \equiv \hat{\theta}_k$.
$\hat{\theta}_X, \theta_{min}, \theta_{max}$
Second stage
Continue to learn by complementing the former Grover measurements with additional ones for disambiguation.
static
for k in {k_X,..., k_{end}}:
Prepare and measure Grover circuits, with r_k increasing exponentially with k.
Sum a constant to r_k. Prepare and measure extra Grover circuits with this factor.
Use angle sum formulas, the knowledge from the first stage $\hat{\theta}_X$ and the latest confidence interval to invert the observed probabilities, getting $\hat{\theta}_k$.
Use Chernoff's inequality to establish a confidence interval around $\hat{\theta}_k$.
$k = k_{end}$?
no
yes
Output $\hat{\theta} \equiv \hat{\theta}_k$; $\hat{a} = \sin^2 \hat{\theta}$.

Figure 9: Diagram summarizing the quantum amplitude estimation algorithm of [42], which relies on a pre-determined exponential amplification schedule. For weak amplification, the observed relative frequencies can be inverted directly to get the Grover angle, which is what happens in the first stage. Note that the applicability of this simple tecnique can be guaranteed in the first iteration by an initial rotation to reduces the Grover angle. The uncertainty due to binomial noise can be bounded using Chernoff's inequality, giving a confidence interval centered at the estimate. Once this range on θ combines with the pre-defined amplification factor for the next iteration to yield a non-injective domain for the likelihood, this simple approach ceases to be valid, as the squared sine likelihood is not invertible for an argument $r_k\theta > \pi/2$. Hence, a passage to stage two is triggered. Solving this ambiguity requires some complementary information. The authors use three sources of information for this purpose: measurements on a supplementary circuit, the resulting estimate of the first stage, and the previous iterations' confidence interval. Note that the algorithm may be interrupted before the first stage finishes if the maximum iteration number is reached, but we omit that case for simplicity.

number of samples, as long as the distribution covers all regions where $f(x)$ is non-null. The simplest way to satisfy that in the general case is to choose a flat distribution, with $\{x_i\}_{i=0}^{N-1} \sim \text{uniform}(\Omega)$. We then have $\pi(x) = 1/V_\Omega$, with V_Ω the volume of the integration domain. In that case:

$$\mathbb{E}_{p(x)}[f(x)] \approx \sum_{i=0}^{N-1} f(x_i)\frac{V_\Omega}{N}p(x_i) \approx \sum_{i=0}^{N-1} f(x_i) \cdot w(x_i), \tag{2.40}$$

where

$$w(x_i) \equiv \frac{p(x_i)}{\sum_i p(x_i)} \tag{2.41}$$

are the weights that arise by representing $p(x)$ as a discretized probability distribution defined to be non-null at the selected points $\{x_i\}_{i=0}^{N-1}$. In other words, we work with a "discretization-tailored" probability mass function. By defining $w(x_i)$ in the discrete domain $\{x_i\}_{i=0}^{N-1}$ with normalized weights, we automatically take care of the V_Ω/N factor associated with uniform sampling: when dividing by normalization constant, which is itself a Monte Carlo estimate, we divide by this same factor.

$$\int p(x)\mathrm{d}x \approx \frac{V_\Omega}{N}\sum_{i=0}^{N-1} p(x_i) \leftrightarrow 1 \approx \frac{V_\Omega}{N}\sum_{i=0}^{N-1} p(x_i) \leftrightarrow \sum_{i=0}^{N-1} p(x_i) \approx \frac{N}{V_\Omega}. \tag{2.42}$$

As an example, we can imagine that $p(x)$ is itself a uniform distribution. This produces the weights:

$$w(x_i) = \frac{1}{N} \text{ for } i \in \{0, \ldots, N-1\} \tag{2.43}$$

for the predetermined samples $\{x_i\}_{i=0}^{N-1}$ from Ω, and $w(x) = 0$ elsewhere. Note that if considering a continuous uniform probability distribution, we'd have instead:

$$p(x) = \frac{1}{V_\Omega} \text{ for } x \in \Omega, \tag{2.44}$$

where V_Ω is the volume of the integration domain. Accordingly, the expectation estimate would become:

$$\mathbb{E}_{p(x)}[f(x)] \approx \frac{V_\Omega}{N}\sum_{i=0}^{N-1} f(x_i)p(x_i). \tag{2.45}$$

That is, the expressions differ by the aforementioned constant factor.

Alternatively, the sample set $\{x_i\}_{i=0}^{N-1}$ can be a grid. Taking the 1-d case and following an elementary quadrature rule, we split the integral into N slices of equal width, and approximate contribution of each slice as the area of the slice-width rectangle whose height is the midpoint's image:

$$\int_a^b g(x)\mathrm{d}x \approx (b-a)g(\frac{a+b}{2}) \text{ with } (b-a) \text{ small,} \tag{2.46}$$

$$\int_A^B g(x)\mathrm{d}x \approx \sum_{x=0}^{N-1} \frac{(B-A)}{N} g\left(A + \left(x + \frac{1}{2}\right)\frac{B-A}{N}\right). \tag{2.47}$$

In the case of equation 2.38 and for a domain $\Omega = [0, x_{\text{max}}]$, this simplifies to:

$$\int_0^{x_{\text{max}}} f(x)p(x)\mathrm{d}x \approx \sum_{x=0}^{N-1} \frac{x_{\text{max}}}{N} p\left(\left(x+\frac{1}{2}\right)\frac{x_{\text{max}}}{N}\right) f\left(\left(x+\frac{1}{2}\right)\frac{x_{\text{max}}}{N}\right). \tag{2.48}$$

Since we can discretize $p(x)$ however we want, we can place the probability mass at simpler locations, namely at the sum indices x. Clearly this does not change anything; f's arguments remain unchanged, and so too does the sum. Additionally absorbing x_{max}/N into $w(x)$ as before, we get:

$$\int_0^{x_{\text{max}}} f(x)p(x)\mathrm{d}x \approx \sum_{x=0}^{N-1} f\left(\left(x+\frac{1}{2}\right)\frac{x_{\text{max}}}{N}\right) \cdot w(x). \tag{2.49}$$

This is the classical approach presented by [56]. Being deterministic, it is not very much in the spirit of Monte Carlo methods; but this rationale helps understand the quantum approach, which we will now present. The use of a quantum algorithm will naturally introduce randomness due to its stochastic nature.

We impose $0 \leq f(x) \leq 1$, because $f(x)$ will be encoded as an amplitude. The algorithm requires an n-qubit register holding the $N = 2^n$ function intputs (x's). A superposition in this register represents a (discretized) probability distribution over x. We introduce an operator P which loads a probability distribution onto this register:

$$P \mid 0 \rangle^{\otimes n} = \sum_{x=0}^{2^n-1} \sqrt{p(x)} \mid x \rangle . \tag{2.50}$$

For instance, a uniform probability mass function can be written into the register by a Hadamard transform. Additionally, for the $f(x)$ evaluations, we introduce an ancilla qubit that will be rotated according to the image of the first register under f.

$$R \mid x \rangle \mid 0 \rangle = \mid x \rangle \left(\sqrt{f(x)} \mid 1 \rangle + \sqrt{1 - f(x)} \mid 0 \rangle \right) . \tag{2.51}$$

The combined action of these two operators is then:

$$R(P \otimes I_1) \mid 0 \rangle^{\otimes n} = \sum_{x=0}^{2^n-1} \sqrt{p(x)} \mid x \rangle \left(\sqrt{f(x)} \mid 1 \rangle + \sqrt{1 - f(x)} \mid 0 \rangle \right) . \tag{2.52}$$

We can rewrite this in terms of two orthonormal subspaces: one in which the ancilla qubit is in state $\mid 1 \rangle$ (first term), and one in which it is in state $\mid 0 \rangle$ (second term).

$$\mid \psi_1 \rangle = \sum_{x=0}^{2^n-1} \sqrt{p(x)} \sqrt{f(x)} \mid x \rangle \mid 1 \rangle . \tag{2.53}$$

$$\mid \psi_0 \rangle = \sum_{x=0}^{2^n-1} \sqrt{p(x)} \sqrt{1 - f(x)} \mid x \rangle \mid 0 \rangle . \tag{2.54}$$

The norm of each of these states is the square root of the sum of the squared superposition coefficients determined by $p(x)$ and $f(x)$:

$$\| \mid \psi_1 \rangle \| = \sqrt{\langle \psi_1 \mid \psi_1 \rangle} = \sqrt{\sum_{x=0}^{2^n-1} p(x) f(x)} \equiv \sqrt{a}. \tag{2.55}$$

and

$$\| \mid \psi_0 \rangle \| = \sqrt{\sum_{x=0}^{2^n-1} p(x)(1 - f(x))} = \sqrt{\sum_{x=0}^{2^n-1} p(x) - \sum_{x=0}^{2^n-1} p(x) f(x)} = \sqrt{1 - \sum_{x=0}^{2^n-1} p(x) f(x)} = \sqrt{1 - a}, \tag{2.56}$$

where we have defined

$$a \equiv \sum_x p(x) f(x) = \mathbb{E}_{p(x)}[f(x)]. \tag{2.57}$$

Using their normalized versions $\left| \tilde{\psi_1} \right\rangle = \frac{1}{\sqrt{a}} \mid \psi_1 \rangle$ and $\left| \tilde{\psi_0} \right\rangle = \frac{1}{\sqrt{1-a}} \mid \psi_0 \rangle$, we can write

$$\mid \psi \rangle = \mid \psi_1 \rangle + \mid \psi_0 \rangle = \sqrt{a} \left| \tilde{\psi_1} \right\rangle + \sqrt{1-a} \left| \tilde{\psi_0} \right\rangle. \tag{2.58}$$

This formulates the expectation computation as quantum amplitude estimation, because a is the expectation value and simultaneously an "amplitude" for QAE to estimate (or rather, the probability of measuring the distinguished state $\left| \tilde{\psi_1} \right\rangle$). The operator A of before consists of the probability and rotation operators:

$$A \equiv R(P \otimes I_1), \tag{2.59}$$

whereas the oracle, which must reflect the distinguished state $\left| \tilde{\psi_1} \right\rangle$, is just a Z gate applied on the ancilla qubit.

$$S_w = (I_n \otimes Z). \tag{2.60}$$

The natural application of this set up is to numerically compute the integral of a squared sine. For instance,

$$I = \frac{1}{b_{\text{max}}} \int_0^{b_{\text{max}}} \sin^2(x) \mathrm{d}x, \tag{2.61}$$

where $p(x) \equiv 1/b_{\text{max}}$ is uniform on the integration domain. By discretizing it on an n point grid and attributing uniform weights ($p(x) = \frac{1}{2^n}$), we can get an expression of the form of equation 2.49:

$$\sum_{x=0}^{2^n-1} \sin^2 \left(\frac{(x + \frac{1}{2}) b_{\text{max}}}{2^n} \right) \cdot \frac{1}{2^n}. \tag{2.62}$$

In this case, the corresponding rotation operator is straightforward to implement. It consists of a single qubit rotation that depends on x:

$$R \mid x \rangle \mid 0 \rangle = \mid x \rangle \left(\sin \left(\frac{(x + \frac{1}{2}) b_{\text{max}}}{2^n} \right) \mid 1 \rangle + \cos \left(\frac{(x + \frac{1}{2}) b_{\text{max}}}{2^n} \right) \mid 0 \rangle \right). \tag{2.63}$$

We now want to write this in terms of familiar 1- and 2-qubit gates. As a building block, we can use rotations of the ancilla qubit around the y axis:

$$R_y(\alpha) \mid 0 \rangle = \sin(\alpha/2) \mid 1 \rangle + \cos(\alpha/2) \mid 0 \rangle. \tag{2.64}$$

Since the effect of a sequence of rotations is to rotate by the sum of the individual rotation angles, we can decompose the operation by rewriting the argument as a sum. Following the ordering convention of [Suzu20] (little endian counting from 1), $x = \sum_{k=1}^{n} x_k 2^{k-1}$. Hence we have:

$$\frac{(\frac{1}{2}+x)b_{\text{max}}}{2^{n-1}}/2 = \frac{b_{\text{max}}}{2^n}/2 + \frac{\sum_{k=1}^{n} b_{\text{max}} \cdot x_k 2^{k-1}}{2^{n-1}}/2 = \frac{b_{\text{max}}}{2^n}/2 + \sum_{k=1}^{n} \frac{b_{\text{max}} \cdot x_k}{2^{n-k}}/2, \tag{2.65}$$

where a $1/2$ constant has been factorized to isolate the arguments for the y rotation gate (equation 2.64). The constant term can be achieved by a single qubit rotation gate acting on the ancilla, and the remaining ones by 2-qubit rotation gates controlled on each of the first register's qubits. In the latter case, the rotation angles are determined by the control qubit's index k. More specifically, they depend on x_k (the k^{th} digit in the bit string x encoded in the quantum state), and on the weight 2^{n-k} (which, having chosen an ordering convention, is determined by the qubit's position in the binary word).

Thus, we can decompose the rotation operator R into the following sequence of gates:

$$R = \prod_{k=1}^{n} \mathrm{C}^{(k)} R_y \left(\frac{b_{\text{max}} \cdot x_k}{2^{n-k}} \right) \left(I_n \otimes R_y \left(\frac{b_{\text{max}}}{2^n} \right) \right), \tag{2.66}$$

$\mathrm{C}^{(k)} R_y$ being the y rotation gate controlled by the qubit with index k and acting on the ancilla. This concludes our example of how to implement a rotation gate in a simple case.

The key takeaway is that **Monte Carlo integration amounts to an amplitude estimation task**. As such, it can benefit from the quadratic speed-up of QAE; this speed-up is in terms of query complexity, and as compared with the classical case. Expectably, the realization of this speed-up faces the same challenges as QAE; and additionally, some other ones.

In particular, loading probability distributions onto a quantum state (i.e. implementing the P operator) may not be easy - or possible - to do efficiently, and may erase the quantum advantage [25]. What is more, the encoding of $f(x)$ itself poses a challenge, requiring arithmetic operations that are exacting for near-term quantum devices [24]. It is important not to disregard these costs, despite their not affecting the oft-quoted quadratic speed-up. This speed-up exclusively considers query complexity, which may be a naive performance metric.

Another pertinent question is how the quantum-enhanced methods compare with *general* integration strategies, and whether they are still of interest when options other than Monte Carlo are considered. Ultimately, to ask this is to question whether Monte Carlo remains a relevant approach to numerical integration. Given its widespread use and (relatedly) its attractive features - generality, simplicity, and dimension-independence -, it is hard to argue otherwise [5]. What is more, its main drawback is the computational cost; thus, a strategy capable of easing this cost while preserving the aforementioned attractive features would be of great interest.

3

Current work and methodology

3.1 Quantum-enhanced estimation

Metrology is the science of measurement; it seeks to learn about real world parameters as efficiently as possible. Should we want to work a regime where quantum-mechanical effects are appreciable, we must use quantum-mechanical tools to arrive at the best reachable performance.

In the case of amplitude estimation, this also applies; once we have encoded the problem in a quantum state, it is not possible to classically surpass the richness of the best quantum data.

These considerations can be summarized in a very simple expressions in terms of the estimation error and the resource count, to be called ϵ and N respectively. The former is the root mean squared error, which is akin to the uncertainty (and can be estimated by the standard deviation). The latter can quantify a number of measurements/probes, a number of queries, an evolution time, etc.

Classically, the fundamental limit for the behavior of the error under an optimal strategy is given by the standard quantum limit (SQL):

$$\boxed{\epsilon^{(SQL)} \propto \frac{1}{\sqrt{N}}} \tag{3.1}$$

This limit is due to statistical noise: the resolution of any measurement is limited by shot noise. For this reason, this limit is also sometimes called the shot noise limit.

If our measurements are independent, this is the best we can hope for. However, when characterizing a system whose quantum mechanical nature is evident, we can exploit quantum effects to improve upon equation 3.1. A correlation between measurements can be introduced via entanglement and/or adaptivity; if put to good use, they can upgrade our estimation results.

More specifically, availing ourselves to quantum state preparation and measurement, we can attain the Heisenberg limit (HL):

$$\boxed{\epsilon^{(HL)} \propto \frac{1}{N}} \tag{3.2}$$

Equation 3.2 represents the ultimate bound of metrology, and the gold standard for estimation tasks.

On noisy quantum devices as we have today, the quantum advantage may not be so clear. Not only are we limited in what we can do, but we also have experimental artifacts tainting the information we collect. This raises an interesting question: how much of this advantage can be maintained given these obstacles?

As it stands, we will find that the answer is rather humbling. This is especially so if we do not accommodating towards the flaws of today's hardware. On the upside, we may be able to extract some benefits from it if we adapt our strategy to its limitations. Even if we do not get as far as Heisenberg limited estimation, we may still observe quantum-enhanced estimation, in the sense that we surpass the standard quantum limit. This is already interesting as a proof of concept, because it shows the little bit of quantumness in our strategy is working its magic.

3.2 Performance benchmarking

In the previous section, we discussed the fundamental advantage in estimation brought about by quantum control. When testing *amplitude* estimation, we will want to keep it in mind. It is the first thing we should look at when assessing the merit of any strategy: where its results fall in relative to the fundamental limits of metrology.

For a graphical depiction of this assessment, we will represent the root mean squared error (root mean squared error) as a function of the number of queries N_q. These queries could be applications of the oracle as a whole, or of the initialization operator. Each of these quantifications is adopted by some authors, with some preferences depending on the application.

For instance, when considering Monte Carlo integration as a purpose, it makes sense to consider the second option, as it tends to be the most complex operator involved. This is what we will opt for.

Either way, the queries to these two operators are nearly proportional, with the initialization operator A being applied twice within each oracle application (once forwards and once backwards, i.e. the inverse) and once more for state preparation.

$$N_{\mathrm{A}} = 2 \cdot N_{\text{oracle}} + 1 \tag{3.3}$$

In light of this, choosing one over the other should not affect our judgement of the algorithms' performance. The most significant difference occurs for the classical case, which never makes any queries to the oracle and so could have a query count of zero in one of the cases. In this sense, it may be wiser to consider queries to the initialization operator.

This ends the discussion on the quantities to be plotted. We now discuss *how* to plot them. Firstly, we will use a double logarithmic scale. This represents the limits in equations 3.1 and 3.2 as straight lines with -0.5 and -1 slopes respectively, facilitating visual assessments.

Secondly, we must decide how to define the y-intercept. This is not given by the limits of the previous section, which describe learning rates only. Hence, to define the vertical offset of the graphs, we used the following strategy:

- Do a curve fit Be^m with parameters B, m:

$$y = Bx^m \leftrightarrow \log(y) = m\log(x) + \log(B)$$
$$(B = e^b)$$

- Calculate the image y_0 of the the x-coordinate x_0 of the first datapoint, under the model from the previous point;

- Make the straight lines of the standard quantum and Heisenberg limits pass by this point.

This seems to produce sensible results, and customizing the y intercept to the datasets facilitates the visual analysis. Of course, the vertical scale offset should still be heeded: it is a relevant indicator of the practical worth of an algorithm.

It is undesirable to have algorithms requiring an excessive resource count from the beginning, even if the reduction of the error then proves to be Heisenberg-limited. Eventually, the performance of such an algorithm will beat that of another with a smaller offset but a less favorable evolution pace. Yet, if the difference is too large, this *eventually* may never arrive. Whether it does depends on how long we run the algorithm, which usually depends on the target resolution.

Another challenge arises from the use of adaptivity. In quantum metrology, the performance metrics tend to respect *average* results, due to the always present statistical noise. These averages are usually taken over algorithm executions, which is straightforward to do. Since these executions always use the same experiment controls and thus the same sequence of query numbers, we simply take the average of all the estimation errors for each query number. For any query number, we are sure to have as many summands as there are runs.

Once adaptivity is introduced, this is no longer as simple: the query numbers are bound to vary between runs. What is more, they tend to increase exponentially, which means that the data will become exponentially sparse as the iteration number grows. Owing to this, it would be extremely difficult to obtain averages by brute force. That is, we could run many simulations and hope that for many distinct and spaced out numbers of queries, there exist coincidental occurrences in a significant proportion of these runs. We could then represent the data points corresponding to these lucky query numbers for which we have sufficient information. However, this is extremely inefficient, especially for large iteration numbers.

Can we do better? Ideally, we would make use of all the data, condensing it into a handful of representative data points. The aim is for the resulting graph to relay the underlying tendency correctly and intelligibly.

A sensible first step is to bin the data, splitting them into groups depending on their numbers of queries. The remaining question is how to process each subset of data to get a single point representing the bin in a way that does not over or under-estimate the learning rate, nor introduce artifacts that hinder visual analysis.

To test possible approaches without having to actually run simulations, which require longer processing and may introduce confounding factors, we generated dummy data observing the HL. We did so as follows:

- Sample an x (representing N) coordinate uniformly at random on a log scale.

$$x \sim \text{unif}_{\log}([x_{\text{min}}, x_{\text{max}}]) \tag{3.4}$$

- Sample auxiliary variables z depending on x:

$$z \sim \mathcal{N}(\mu, \sigma(x)), \tag{3.5}$$

  where

$$\sigma(x) \propto 1/x. \tag{3.6}$$

  The choice of the mean μ is arbitrary, whereas the constant of proportionality is determined by fixing a point.

- Calculate y (representing ϵ^2) as $y = (x - \mu)^2$.

Clearly, points thus generated will emulate ideal Heisenberg limited behavior:

$$\mathbb{E}(y) = \mathbb{V}(x) \propto 1/x^2 \tag{3.7}$$

The goal is to find a processing strategy that will portray them accordingly.

Alternatively to steps 2-3, we could assign $y = \sigma(x)$ to get "noiseless" points standing directly on the HL. This does not mimic the behaviour of actual Heisenberg limited estimation, to which randomness is intrinsic. It is merely meant to aid the reasoning.

A plot of data generated like so should resemble figure 10.

Now, if instead of each bin containing scattered points we had multiple points with the same x, we could simply average their y coordinates to get a well-behaved representation. However, since the x coordinate is distributed on a continuum, only with vanishing probability will any two points get the same x.

In this case, the naive approach would be to average their x and y coordinates independently, but this does not seem justified. We need only think of points lying on a straight line. Clearly, averaging over y for constant x will produce another point on the line; but averaging over each coordinate will not.

We tried subjecting the groups (datapoints in each bin) to the following treatments:

1. **Independent x/y averaging.** Average their x coordinates and their y coordinates separately - the naive approach. After averaging, take the square root of $\bar{y}$ to get a quantity akin to the estimation error (RMSE, or in this case standard deviation). Plot $\bar{x}$, $\sqrt{\bar{y}}$.

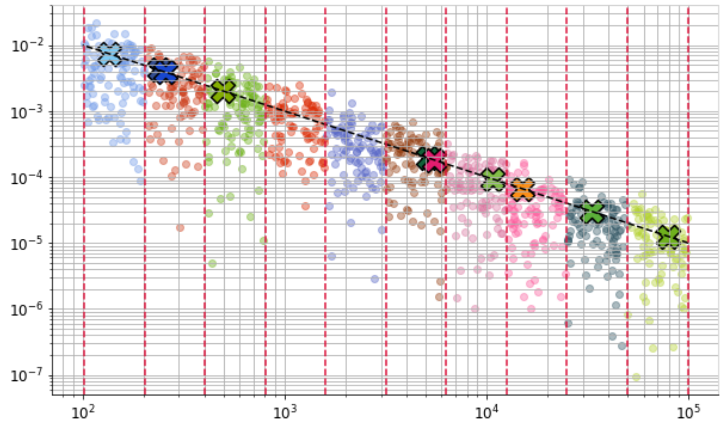


Figure 10: Graphical representation of the "Heisenberg-limited" dummy data generated as described in the text (colorful dots). The data have been divided into bins, whose frontiers are represented by dashed vertical lines. Points in the same bin are plotted in the same color. The 'x' markers represent "noiseless" points, which lie on the HL (dashed diagonal line). They are fewer so as not to clutter the graph.

2. **Average "log-"coordinates.** Same as the previous point, but doing the averages in the log-domain, i.e. taking logarithms before averaging, then exponentiating the averages.

3. **Average slopes.** Average their x coordinates to get x. Separately, computing the (log-)slope of each point in the group relative to the fixed point, and take the average slope $\bar{m}$. Compute the image of the mean x under the power function given the average slope, and the fixed point: $f_{\bar{m}}(\bar{x})$.

The reason for trying the two latter options is the intuition that in a noiseless case the processed points should all exactly overlapped with a line, as do the original ones. The obvious way to fix it is to work in the log domain, since an affine function $f(x) = mx+b$ applied to an average, $f(\bar{x})$, is the average of the images of the individual elements, $\overline{f(x)}$. E.g., $f((x_1+x_2)/2) = m(x_1+x_2)/2+b = [(mx_1+b)+(mx_2+b)]/2$.

We further tried two strategies without prior binning:

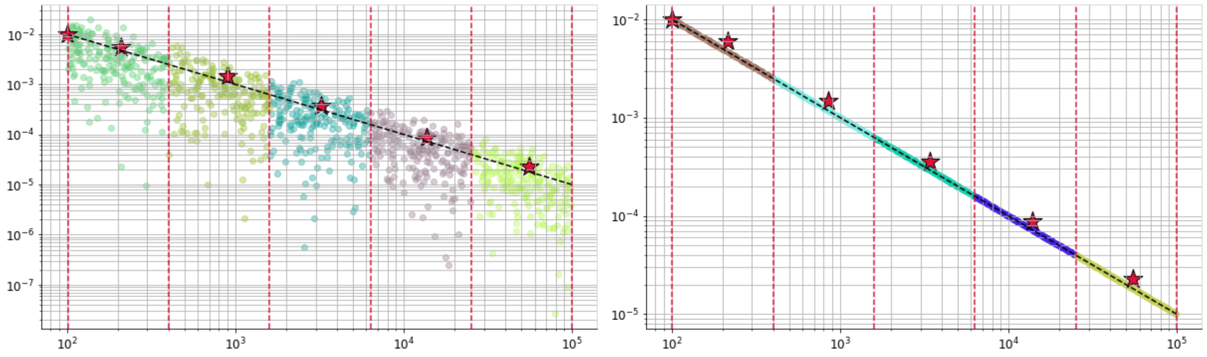
5. **Curve fitting**.

6. **Spline interpolation**.

The motivation for these is clear: they are standard data treatment strategies. In this case the results are functions, which can then be evaluated at the bin midpoints to produce evenly spaced sample points comparable to those of the other strategies.

Figures 11 to 15 showcase the performance of each of these approaches. An ideal candidate is one who translates each group of binned data points into a single point lying on the dashed line.

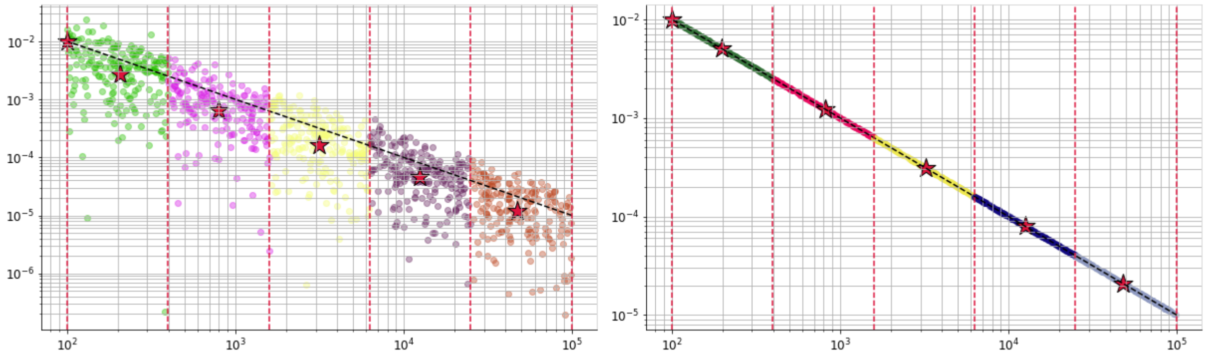
Interestingly, the method that seems to work best for our purposes is the naive one - despite the fact that most other methods work near perfectly with the noiseless dataset, which would intuitively seem like a good property. When operating over actual points whose images are probabilistic, they seem to introduce bias.



(a) Noisy case.

(b) Noiseless case.

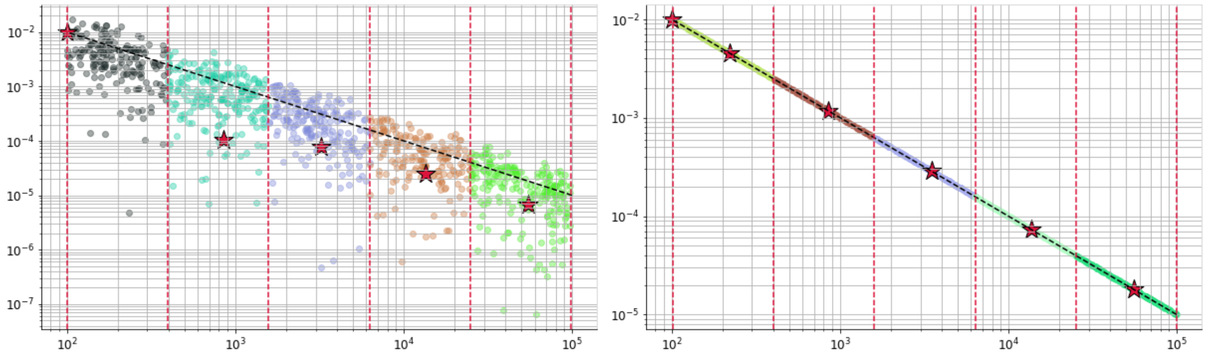
Figure 11: Summary data points obtained by strategy 1 - independent x, y averaging.



(a) Noisy case.

(b) Noiseless case.

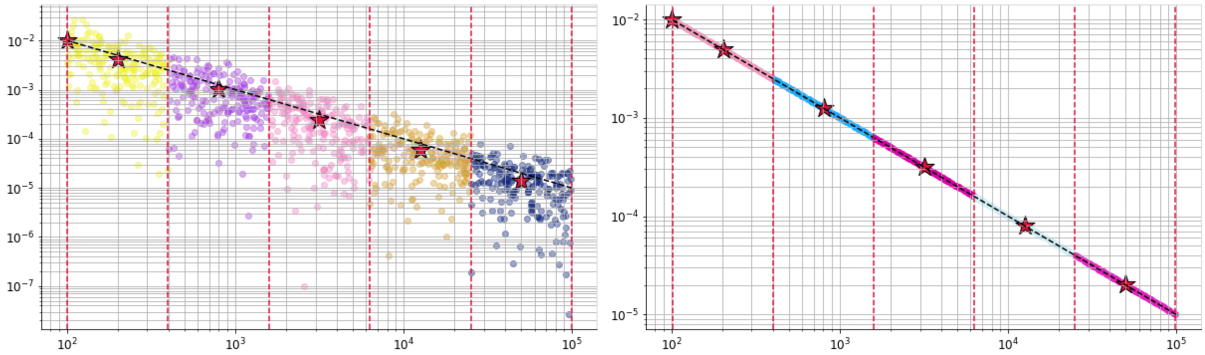
Figure 12: Summary data points obtained by strategy 2 - average ”log-”coordinates $\log(x)$, $\log(y)$.



(a) Noisy case.

(b) Noiseless case.

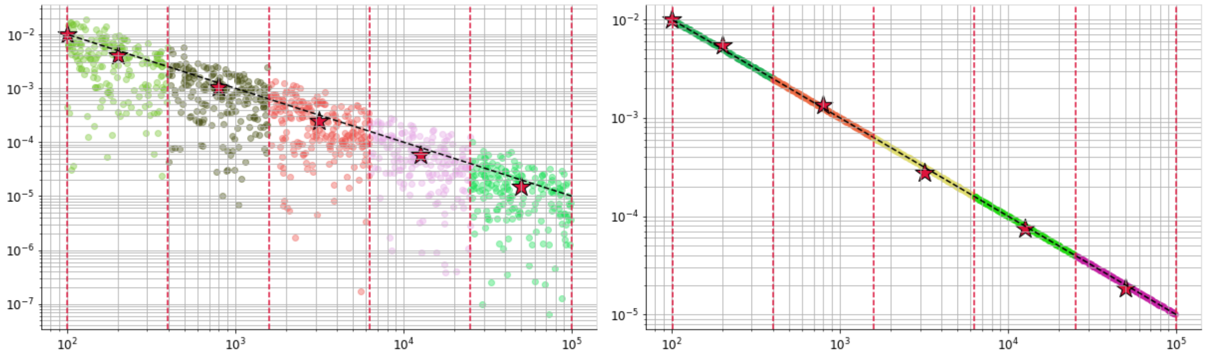
Figure 13: Summary data points obtained by strategy 3 - average slopes.



(a) Noisy case.

(b) Noiseless case.

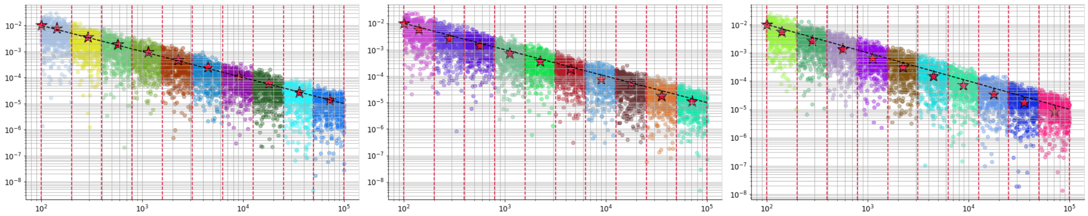
Figure 14: Summary data points obtained by strategy 4 - curve fitting.



(a) Noisy case.

(b) Noiseless case.

Figure 15: Summary data points obtained by strategy 5 - spline interpolation.



(a) Independent x/y averaging.

(b) Spline interpolation.

(c) Curve fitting.

Figure 16: Summary data points obtained by the 3 best performing strategies.

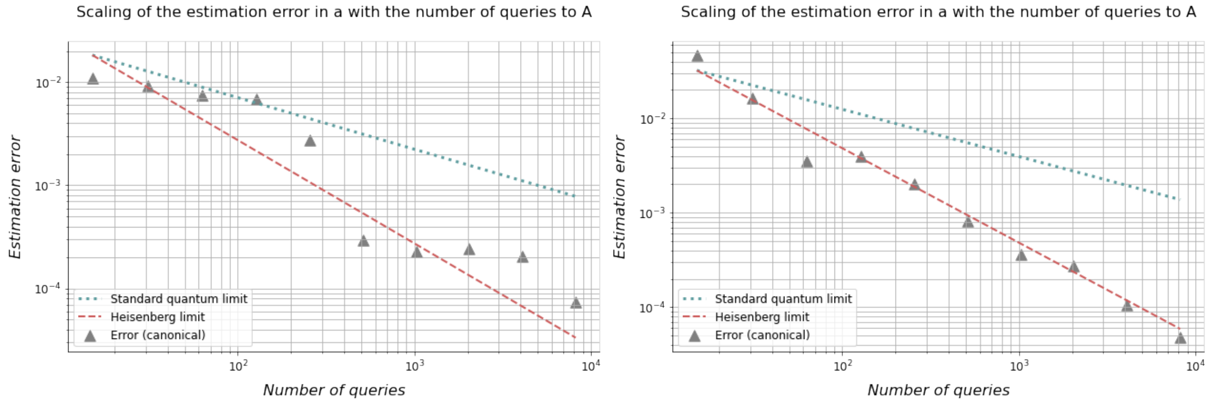
The graphs produced by the top 3 contenders are shown in figure 16. Independent x/y averaging seems to yield the best results; accordingly, we will adopt this strategy where necessary.

3.3 Numerical experiments

In this section we present the results of running simulations for the main algorithms described in chapter 2. Shot noise is always present, but the measurements are otherwise noiseless. The raw data is processed as described in section 3.2; and as therein, the graphs represent how the relevant metric (the root mean squared error) evolves as a function of the key resource (the number of queries N_q). The standard quantum limit (SQL) and HL are also depicted for reference (cf. section 3.1).

Before proceeding, note that we will use a constant value for the real amplitude, here 0.5. This is a common approach, see e.g. [22]. However, in our experiments, we observed that such a choice tends to produce particularly smooth results. While that is interesting in the sense that it is easier to analyse and compare graphs while averaging over less runs, it may set somewhat unrealistic expectations.

Even the canonical algorithm, which is the gold standard, shows stronger irregularities for other choices of a. This is depicted in figure 17, where we had a take an irrational value (figure 17a) and registered marked oscillations around the HL.



(a) Real amplitude $a = 0.1\pi$.

(b) Amplitudes sampled uniformly at random from $[0, 1]$ for each run.

Figure 17: Evolution of the root mean squared error in the amplitude estimate obtained by canonical quantum amplitude estimation [4] with the number of queries. The results were averaged over 10^2 runs, with the data from each run being post-processed using MLE as described in section 2.3.

Perhaps a fairer test would be to pick a fresh amplitude uniformly at random for each run, and average the results. This does produce smoother looking data points, as can be seen in figure 17a - though still not as smooth as a constant and equal to 0.5 would. Both results should be compared to 18b; they each differ from the others only in the choice of a.

Note also that we post-processed the canonical QAE results using MLE as proposed in [22] (refer to section 2.3 for details). This too has a regularizing effect on the results. In particular, applying textook QAE to $a = 0.5$ could never produce a graph as 18b.

This is because the algorithm is exact for angles $\theta = \pi x / 2^m$ with x an integer in $\{0..2^m - 1\}$ (the measured binary outcome as an integer). In $\theta \in [0, \pi/2]$, $a = 0.5$ corresponds to an angle $\theta = \pi/4$, which is representable whenever 2^{m-2} is an integer (so x can be $= 2^m - 2$). This happens for any $m \geq 2$.

In particular, QAE is exact for $a = 0.5$ as long as the number of qubits is larger than 1. As such, this may not be a fair test case for performance, there being infinitely more cases where QAE can't yield a finite solution. It is also one of the few cases where MLE does not improve upon canonical QAE; rather, it requires extra processing while making the results worse. With the canonical algorithm, the error is 0

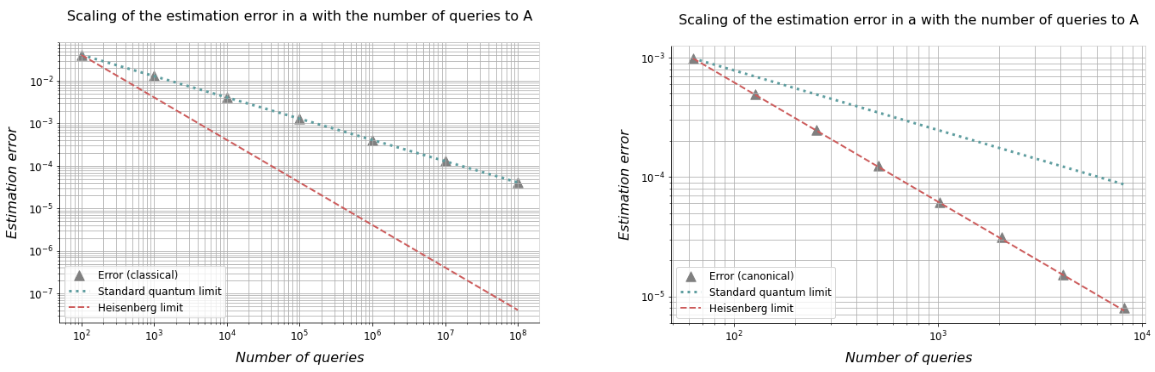
as soon as the number of qubits crosses 1, for whichever number of shots. In the case of MLE, there is always an uncertainty associated with any estimator, since it is always possible that other parameters (for which QAE is not exact) produced the outcome corresponding to some exactly representable amplitude.

In any case, the impact is less drastic on other algorithms, which tend not to be so partisan with respect to the values a. As for the canonical algorithm, its performance is well studied, and not the goal of our studies. Thus, having made this aside on the bias our choices may entail, we proceed to present the numerical results.

Figure 18a shows the best possible performance for a classical algorithm, achieved by drawing N samples from a binomial distribution with parameter a and taking the fraction of successes (measuring a marked state) as an estimate for a. The number of queries is equal to the number of samples.

$$S \sim B(N, a) \tag{3.8}$$

$$a \approx S/N \tag{3.9}$$



(a) Results obtained by classical amplitude estimation (sample means).

(b) Results obtained obtained by canonical quantum amplitude estimation [4]. The data from each run was post-processed using MLE as described in section 2.3.

Figure 18: Performance of the best classical and quantum algorithms for amplitude estimation. The graphs show the evolution of the root mean squared error in the amplitude estimate with the number of queries. The real amplitude is $a = 0.5$ and the results were averaged over 10^2 runs.

In this case, the optimal behavior is given by the standard quantum limit. The estimation error shrinks as $O(\sqrt{N})$ with N the number of queries.

On the other hand, figure 18b shows the best possible performance for a *quantum* algorithm, which is given by the HL. It was obtained using the canonical QAE algorithm of section 2.2. The estimation error shrinks as $O(N)$ with N the number of queries.

The following graph respects the maximum likelihood amplitude estimation algorithm of [56], which is more irregular than the previous two. Results are presented for two different strategies: in one of them the Grover exponent is increased linearly with the iteration number, and in the other it grows exponentially.

The authors call these strategies LIS and exponentially incremental sequence (EIS) respectively. The latter performs better, coming closer to the HL; the former falls approximately midway between the Heisenberg and standard quantum limit. Both are clearly quantum-enhanced, with sub-SQL performances.

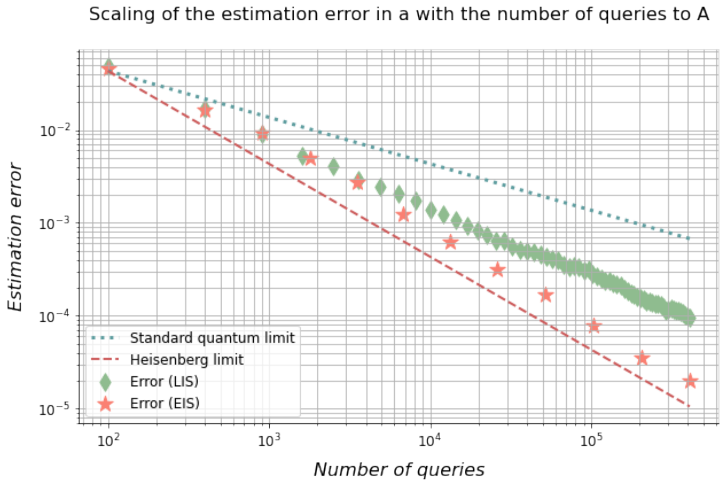


Figure 19: Evolution of the root mean squared error in the amplitude estimate obtained by maximum likelihood quantum amplitude estimation [56] with the number of queries. Here the real amplitude is $a = 0.5$, and the results were averaged over 10^2 runs. The algorithm used 100 shots per circuit, and both exponentially and linearly increasing strategies were tested for the Grover circuit schedules.

The performance for MLAE is limited at best by the **lower bounds** derived in [56]:

$$N_q^{\text{(LIS)}} \sim \mathcal{O}(\epsilon^{-0.75}) \tag{3.10}$$

$$N_q^{\text{(EIS)}} \sim \mathcal{O}(\epsilon^{-1}) \tag{3.11}$$

This is not a very satisfactory result per se, as it says nothing about the worst or average cases. However, curve fits on the results in 19 give rather interesting **numerical results**:

$$N_q^{\text{(LIS)}} \sim \mathcal{O}(\epsilon^{-0.76}) \tag{3.12}$$

$$N_q^{\text{(EIS)}} \sim \mathcal{O}(\epsilon^{-0.88}) \tag{3.13}$$

This is a remarkable feat for such a simplistic and lightweight algorithm, even though the quantum advantage is not present in full.

Contrastingly, in figure 20, the ”quantum amplitude estimation, simplified” algorithm recovers in full the advantage seen in figure 18b, with the points neatly aligned on the standard quantum limit.

However, the order of magnitude of the number of queries is rather daunting, starting out at a number larger than that of MLAE by roughly a one-million factor. As we will see, no other algorithm has this colossal a cost offset. While the impeccable scaling means that for large enough query counts AES will beat less-than-perfect algorithms, the greatness of the discrepancy makes this unlikely for practical problem instances and sizes. Nevertheless, the theoretical merits of AES are unquestionable, as the first algorithm to achieve ideal query complexity while using nothing but simple quantum amplification circuits.

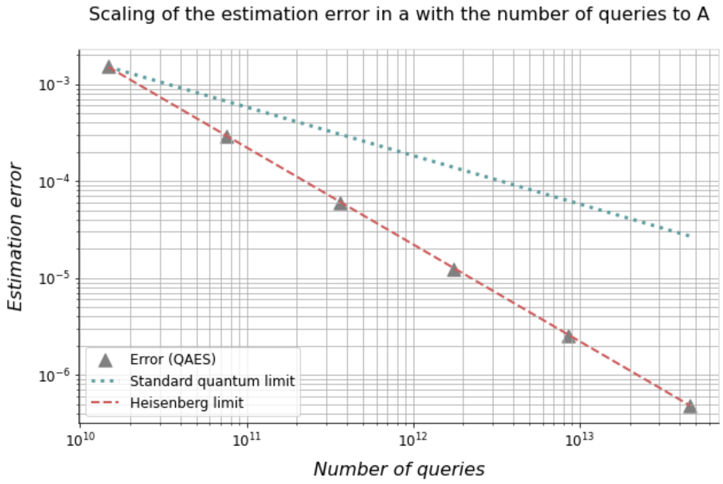


Figure 20: Evolution of the root mean squared error in the amplitude estimate obtained by "quantum amplitude estimation, simplified" [1] with the number of queries. Here the real amplitude is $a = 0.5$, and the results were averaged over 10^2 runs. The input failure probability was $\alpha = 0.001$.

The next algorithm, SAE, does not enjoy the same quality, with the data points in 21 making an erratic pattern that does not convincingly achieve sub shot noise estimation. At any rate, the arguments presented in the paper do not seem to provide convincing guarantees that such would be the case.

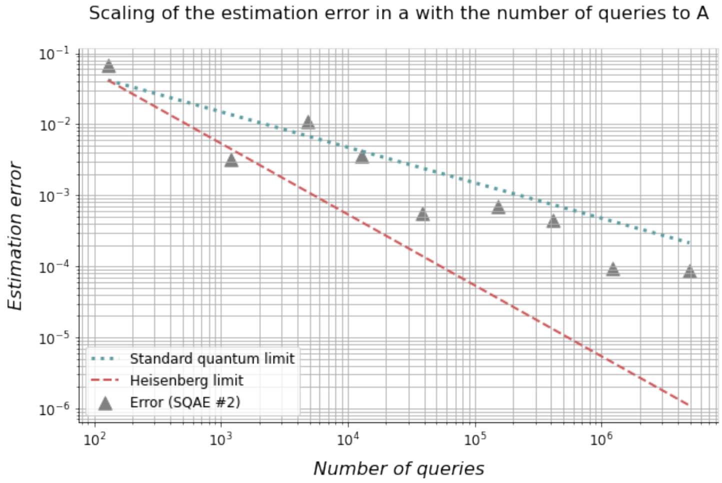


Figure 21: Evolution of the root mean squared error in the amplitude estimate obtained by the simpler quantum amplitude estimation [61] algorithm with the number of queries. Here the real amplitude is $a = 0.5$, the results were averaged over 10^6 runs, and the recursive inversion formula proposed in the original paper was used.

Back to formally rigorous algorithms, iterative amplitude estimation demonstrates its robustness in figure 22. The logarithmic factor does not visibly affect performance.

Finally, faster amplitude estimation also offers reliable Heisenberg-limited estimation, as can be seen in figure 23. Even though the data looks somewhat noisy, the trend is clear.

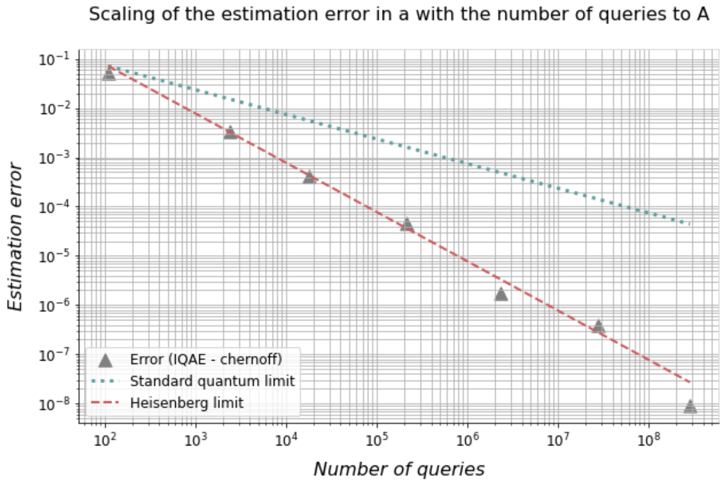


Figure 22: Evolution of the root mean squared error in the amplitude estimate obtained by the iterative quantum amplitude estimation [22] algorithm with the number of queries. Here the real amplitude is $a = 0.5$, and the results were averaged over 10^2 runs. The Chernoff-Hoeffding inequality was used to define the confidence intervals, the input failure probability was $\alpha = 0.05$, and 100 shots were used for each measurement.

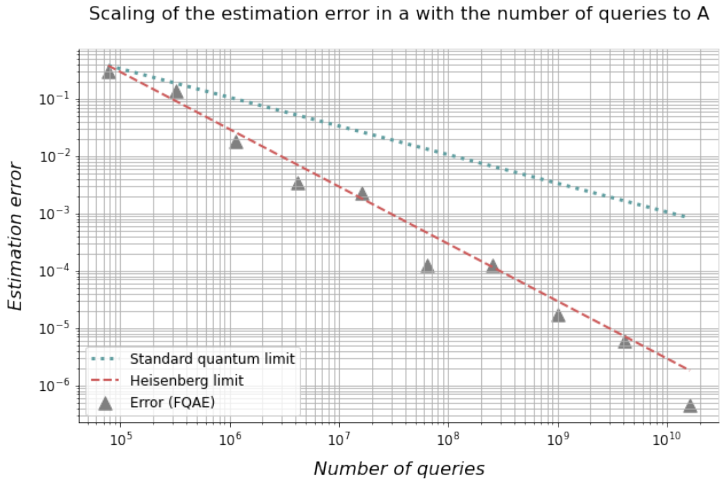


Figure 23: Evolution of the root mean squared error in the amplitude estimate obtained by the faster quantum amplitude estimation [42] algorithm with the number of queries. Here the real amplitude is $a = 0.5$, and the results were averaged over 10^6 runs. The failure probability for estimates in the first stage was set to $\delta_c = 0.01$ (this is an algorithm parameter).

3.3.1 Numerical experiments in the presence of noise

Having assessed the performance of a selection of algorithms in a noise-free setting, we would like to re-assess it under a simple noise model.

Arguably the most relevant type of noise in today's quantum devices is decoherence [43, 12, 18], a phenomenon causing gradual information loss. While the state of an ideal quantum computer would be affected exclusively by the unitary gates acting on it, real devices are also affected by the passage of time. The magnitude of this effect can be relayed by a characteristic time termed *coherence time* which we will denote by T.

In an idealized setting, the coherence time is infinite. In the real world, it is often of the order of microseconds, with depolarization, energy relaxation and dephasing contributing to its minuteness. The precise value depends on the architecture, the specifics of the physical implementation, the environment, and other factors.

In practice, a finite coherence time means that the wavefunction is damped - the smaller the time constant, the more intense the damping. In the spin oscillations literature referred to in section 1.6, this damping is often modelled using an exponential decay envelope [21, 51]:

$$\mathbf{P}(1 \mid t) = \sin^2(\omega t) \xrightarrow{\text{decoherence}} \mathbf{P}(1 \mid t) = e^{-t/T} \sin^2(\omega t) + \frac{1 - e^{-t/T}}{2}. \tag{3.14}$$

Here ω is a frequency and the parameter under study, whereas t is an experimental control (the evolution time).

We can extrapolate this to a generic amplitude estimation circuit using m applications of the Grover operator (plus an initialization step):

$$\mathbf{P}(1 \mid m) = \sin^2((2m+1)\theta) \xrightarrow{\text{decoherence}} \mathbf{P}(1 \mid m) = e^{-m/T} \sin^2((2m+1)\theta) + \frac{1 - e^{-m/T}}{2}. \tag{3.15}$$

In this case, as we know, θ is the Grover angle (the parameter to be learned), and m is an experimental control, and T amounts to a "discrete" coherence time, roughly in units of duration of a single Grover application (up to initialization). Note the dissonance between the exponent and the sinusoid's argument as compared to the previous case. The point is that to prepare the state corresponding to the likelihood $\sin^2((2m+1)\theta)$, m Grover iterations are necessary. Since the depth of the circuit determines the strength of the decoherence[1], this is what situates us in the decoherence timescale.

Simple as they are, these considerations allow us to examine the behavior of amplitude estimation algorithms in the face of strong decoherence, a very relevant issue in today's quantum hardware.

Before proceeding, note that in this case it is particularly unwise to make amplitude taking the value $a = 0.5$, as this is precisely the amplitude associated with the maximally mixed state. Despite this not being the state we start out with, which is pure (by the problem statement), it works identically for amplitude estimation purposes.

We start our numerical tests by the classical amplitude estimation algorithm in figure 24. Unsurprisingly, the impact in its performance is negligible. This is because only the most shallow possible circuit is used, corresponding to an initialization; the outlook is not expected to change even for small coherence times. At the time-scale where this algorithm would be affected, all others would be completely unfeasible.

The next algorithm in the sequence would be the original QFT-based quantum amplitude estimation algorithm. However, the introduction of noise complicates the analytical specification of the outcome distribution, and other simulations would demand impractical run-times. In any case, given the extensive

[1] Along with the physical device characteristics, which we assume to be constant.

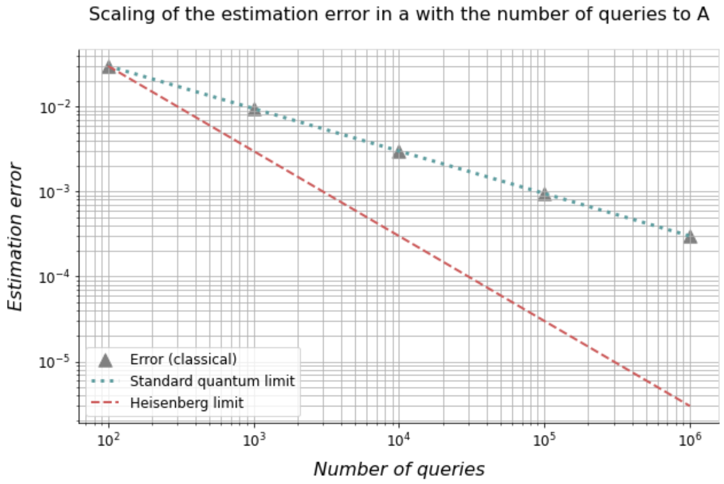


Figure 24: Evolution of the root mean squared error in the amplitude estimate obtained by classical amplitude estimation (sample means) with the number of queries, under the presence of decoherence. Here the real amplitude is $a = 0.1$, the coherence time is $T = 2000$, and the results were averaged over 10^5 runs.

circuit implied and the coherence timescales considered here, it is not hard to imagine what the graph would look like if we were to obtain it: the error would not decrease regardless of the number of queries.

Then, in figure 25, we have maximum likelihood amplitude estimation. We fixed the coherence time at $T = 2000$. There is a noticeable difference in the evolution of the error, with both strategies being deviated from the HL and towards the standard quantum limit. Furthermore, towards the end, the superiority of the EIS strategy relative to LIS is overturned. This is because the latter grows the circuits at a slower pace. As a result, when matching their query counts, LIS relies on shallower circuits, which are less affected by decoherence. This is an interesting example of how algorithm variations that perform better in an ideal setting may not in a noisy one.

The next algorithm in line is quantum amplitude estimation, simplified. However, this algorithm could not produce results at all. This is due to a cycle termination condition enforcing that the amplitude be amplified to over 0.95. Under the presence of decoherence, this constitutes a troublesome demand: unless the amplitude and/or the coherence time are quite large, it may be entirely impossible to satisfy such a condition. This is an interesting example of how an algorithm can rely on shorter circuits, yet be extremely vulnerable to realistic noise sources. It is not only the circuit structure that matters, but also the way the measurement data are incorporated into the steps that the algorithm comprises.

The ensuing algorithm, simpler amplitude estimation, does terminate, but its performance falls short of what would be desirable - as figure 26 demonstrates. Any quantum advantage that might have been present is completely erased after the first couple of data points.

In figure 27, we can see the that iterative amplitude estimation boasts a higher standard, with the quantum advantage enduring for longer despite the more erratic evolution (as compared to figure 22).

Finally, faster amplitude estimation does too show some degree of resistance, with clearly sub-SQL - if not Heisenberg limited - estimation. This can be seen in figure 28.

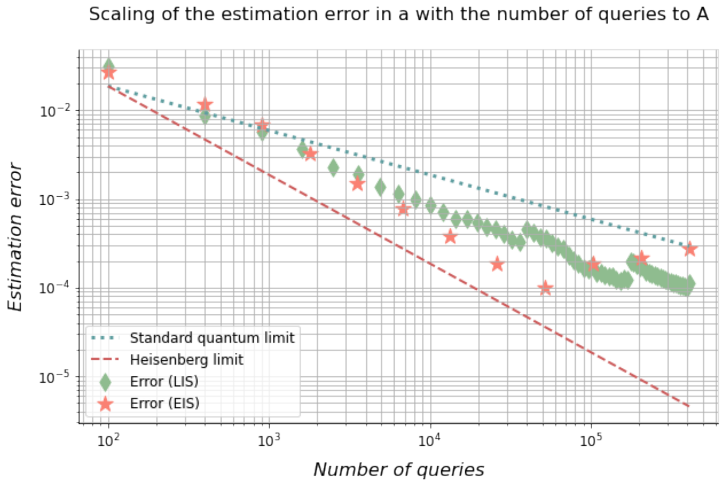


Figure 25: Evolution of the root mean squared error in the amplitude estimate obtained by maximum likelihood quantum amplitude estimation [56] with the number of queries, under the presence of decoherence. Here the real amplitude is $a = 0.1$, the coherence time is $T = 2000$, and the results were averaged over 10^2 runs. The algorithm used 100 shots per circuit, and both exponentially and linearly increasing strategies were tested for the Grover circuit schedules.

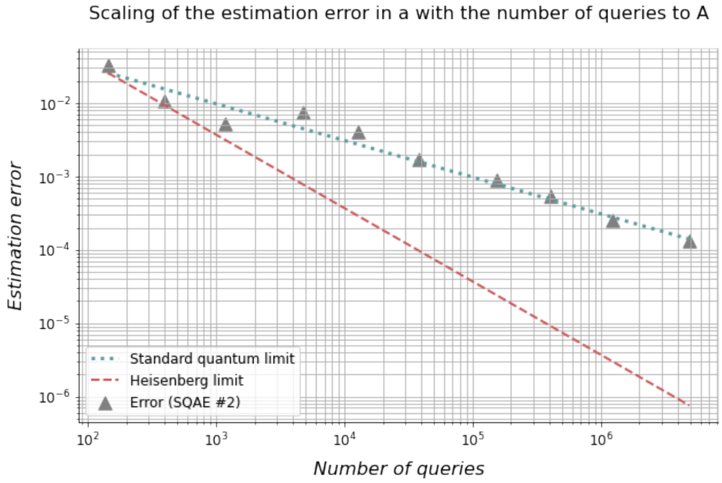


Figure 26: Evolution of the root mean squared error in the amplitude estimate obtained by the simpler quantum amplitude estimation [61] algorithm with the number of queries, under the presence of decoherence. Here the real amplitude is $a = 0.1$, the coherence time is $T = 2000$, and the results were averaged over 10^6 runs. The recursive inversion formula proposed in the original paper was used.

3.4 Bayesian amplitude estimation

An alternative approach to amplitude estimation can come from the application of Bayesian inference. This rather flexible and robust methodology has been popularized in the scientific community, namely for the characterization of quantum systems [27, 60, 6, 28].

This type of inference relies on the systematic application of Bayes' rule:

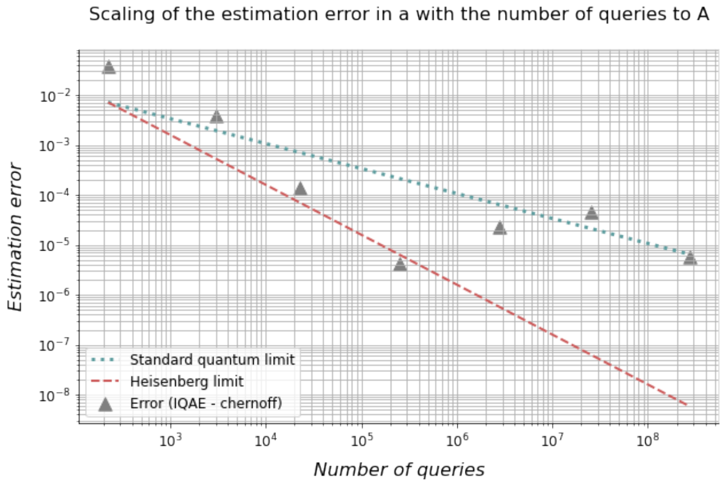


Figure 27: Evolution of the root mean squared error in the amplitude estimate obtained by the iterative quantum amplitude estimation [22] algorithm with the number of queries, under the presence of decoherence. Here the real amplitude is $a = 0.1$, the coherence time is $T = 2000$, and the results were averaged over 10^4 runs. The Chernoff-Hoeffding inequality was used to define the confidence intervals, the input failure probability was $\alpha = 0.05$, and 100 shots were used for each measurement.

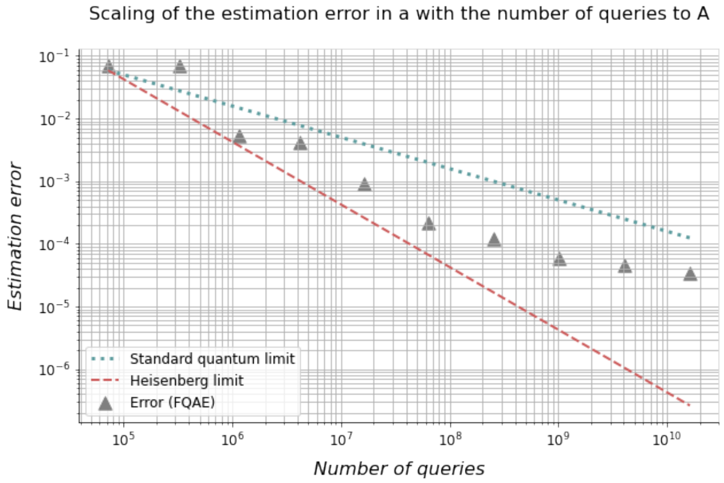


Figure 28: Evolution of the root mean squared error in the amplitude estimate obtained by the iterative quantum amplitude estimation [22] algorithm with the number of queries, under the presence of decoherence. Here the real amplitude is $a = 0.1$, the coherence time is $T = 2000$, and the results were averaged over 10^6 runs. The failure probability for estimates in the first stage was set to $\delta_c = 0.01$ (this is an algorithm parameter).

$$\boxed{\mathbf{P}(\theta \mid D) = \frac{\mathbf{L}(\theta \mid D; E)\mathbf{P}(\theta)}{\mathbf{P}(D; E)}} \tag{3.16}$$

Here θ is the parameter of interest, D is a datum, and E is an experiment. The expression reads as: the probability of θ being the real parameter given that we performed experiment E (in this case, some form of Grover circuit) and obtained outcome $D \in \{0, 1\}$ is given by the likelihood of the parameter given the observation (defined as the probability that the former would have generated the latter), times our prior degree of belief in θ, divided by a normalizing constant.

This expression can be used to extract empirical knowledge from a dataset; the data can be considered sequentially or via a batch update.

Equation 3.16 does not directly give us an estimate of the parameter, but rather a function quantifying the merit of any parameter value. To obtain a numerical single point estimate, we must still do some processing. One possibility is to sweep through the parameter domain and find the mode. Another is to integrate (analytically or numerically) and find the mean.

For the results to be shown here, we opted for the latter, which tends to be more reliable and versatile. In particular, we used a numerical method called Sequential Monte Carlo (or particle filter) [10] with Liu-West kernel shrinkage and smoothing [36]. See [21] for a quick introduction. More sophisticated methods exist [54, 9, 8], but they are also more computationally costly, and do not necessarily bring an advantage for simple low-dimensional cases as this one.

One major benefit of this approach is allowing for processing the data sequentially, as they arrive. At each point of the inference process, a full Monte Carlo description of the current information is available. This can be used to estimate the expected value of any function of the parameters - which includes *utility* functions, such as the uncertainty.

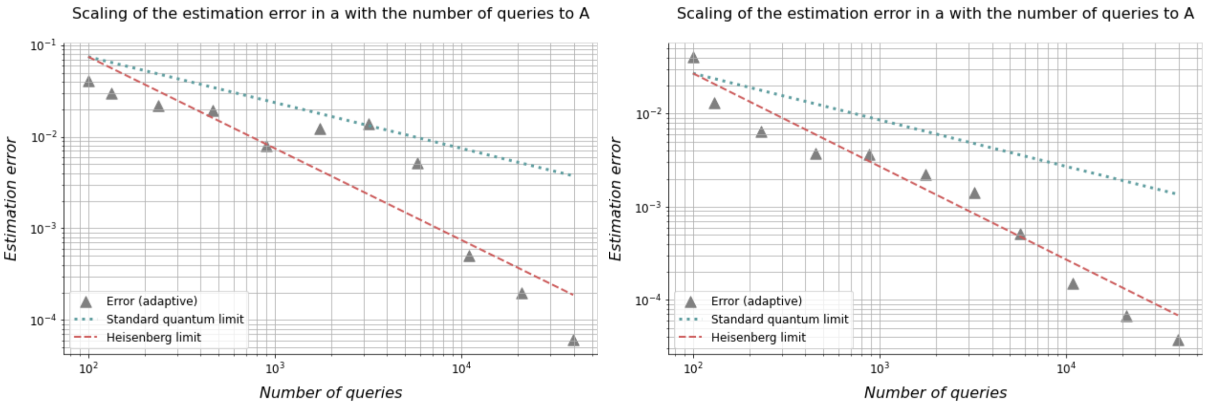
Apart from quantifying our confidence, this is useful because the uncertainty can be calculated even in a *look-ahead* scheme, where we compute the expected utility conditional on some specific experimental controls (such as the Grover experiment). As such, we can sweep over potential experimental controls and choose the ones we expect to produce better results.

The downside is that the operations involved can be quite costly - more the larger the look-ahead. To remedy this, one may opt for a unitary look-ahead, adopting what is usually called a greedy strategy. It is only locally optimal, and does not necessarily produce the best global results, but has been shown to have a good performance [14]. Note that this is an adaptive strategy, requiring online processing. This is the case as long as the look-ahead is shorter than the total number of experiments to be performed.

The results of putting this into practice are shown in figure 29, where we consider no decoherence as in section 3.3. Sub-figure 29a shows the usual quantity, which is the mean squared error. The error seems to follow a roughly Heisenberg-limited tendency, but strong irregularities are present. The *median* squared error, presented in 29b, has a quite smoother evolution. This is indicative of the existence of aberrant runs due to e.g. numerical instability.

In practice, the median is often used, due to being a more robust estimator. In algorithms with a more stable performance, such as the ones presented earlier, the mean and median results are expected to be comparable. But when there are significant inter-run fluctuations, the median helps smooth over irregularities that may be present.

Notice that these considerations do not respect only the analysis of results: they transfer directly to the *post-processing* of results. In real-world applications, we don't have to trust the result of a single execution. In algorithms like this, it is wise to repeat the tests a few times and then take the median estimator. This ensures a more reliable result.



(a) Evolution of the mean estimation error.

(b) Evolution of the median estimation error.

Figure 29: Evolution of the root mean squared error in the amplitude estimate obtained by Bayesian quantum amplitude estimation with the number of queries. Here the real amplitude is $a = 0.1$, and the results were averaged over 125 runs. After a warm-up, the utility was optimized greedily by brute force grid search on $1, \ldots, 1000$. The warm-up consisted of 100 repetitions of the same non-amplified measurement (classical sampling). 1000 particles were used for the Liu-West filter, for which a filtering parameter of 0.98 and a resampling threshold of 0.5 were used.

As compared to others, this algorithm implies a classical processing overhead (for the optimization of the experimental controls). Its advantage resides in the agility afforded by its thorough treatment of the available information: both the model and the experimental strategy can be adapted to perform better in the presence of noise.

In particular, they can easily be made to accommodate noise models, whereas the algorithms seen in section 2.4 cannot: most of them are, just as the original QAE algorithm, reliant on ideal measurements. That those measurements correspond to simpler circuits is an improvement, but an insufficient one.

A Bayesian approach brings, by virtue of its suppleness, the ability to study the undesirable properties alongside the parameters of interest. As the former undermine the characterization of the latter, this offers a largely customizable path to robust characterization. On the one hand, it can account for the fact that the system's behavior is affected by confounding factors. On the other, it can be a stepping stone for realistic measurement selection under less than ideal conditions.

Both Bayesian statistics and quantum noise modelling are immense areas of research, due to their usefulness; thus, many strategies can be explored within this scope, with the overall aim to estimate not only amplitudes, but also whatever is necessary along with them.

3.4.1 Robust amplitude estimation

Techniques similar to the ones described in the previous section have been applied to amplitude estimation in [59], in a strategy termed robust (quantum) amplitude estimation (RAE). This represents a quite more involved framework than other algorithms, but the investment can be rather rewarding.

This proposal relies on Bayesian inference with *engineered likelihood functions*, which are achieved by generalizing the Grover operator; or equivalently, by further customizing the circuit model to be used for data collection. This broadens the range of attainable likelihood functions - a perk which, should it be put to good use, can reduce the runtime of amplitude estimation via an increase in the learning rate. Moreover, a simple noise model is incorporated into the likelihood, which further upskills the experiment selection.

Armed with this flexibility, the authors attempt to greedily maximize the variance reduction by adaptively optimizing proxies thereof. On account of the heavily personalized circuits, the optimization has more degrees of freedom to work with, enabling an especially meticulous experimental design. This is a novelty as compared to other amplitude estimation algorithms, as is the introduction of a model to account for device error throughout the execution.

Instead of numerically approximating the distribution as we did in section 3.4, the authors work under a Gaussian assumption that makes the representation analytically tractable. The downside of such an approach is that the limited expressibility may lead to the inference results' - both intermediate and final - being wrongly captured, skewing the experimental choices and the final estimates in the wrong direction. While this may not be apparent in simple test cases, it becomes so once the models are sufficiently intricate.

Such a scenario is likely to occur when pursuing a thorough noise characterization, the simplicity of QAE's fundamental model notwithstanding. This is due to the fact that capturing noise accurately requires augmenting the model with additional parameters: the higher the dimension of the parameter space, the harder it is to explore, and the more unforgiving towards simplistic methods. For instance, high dimensional spaces are prone to extrinsic redundancy, which undermines the normality assumption that Gaussian approximations rest upon.

This extreme example is only one of several ways an inadequate representation method may compromise the inference efforts. More nuanced oversights are even more likely to arise, misguiding the experimental design and jeopardizing accuracy. Such shortcomings cannot be overcome even by a perfect generative model for the noisy data.

In short, the statistical backbone plays a pivotal role in the inference. It is thus critical that it be refined along with its other aspects, so as not to disrupt the efforts invested in them.

Finally, we note that the proposal of [59] has been expanded upon by others:

- In [31], it is tested on an IBM quantum device, and its noise mitigating abilities assessed.
- In [35], an in depth analysis of the measurement circuit optimization (for the likelihood engineering) is presented; the framework is thoroughly studied, but the noise considerations are forgone.
- [29] assesses how much RAE speeds up energy expectation estimation in VQEs for specific molecules.
- The authors of [7] remark that the exponential decay model considered by others is not realistic, and attempt to make it so by changing the noise to fit it. For that, they use randomized compiling, which transforms coherent errors into stochastic, easier to characterize ones.

4

Future work

4.1 Research directions

We now would like to present the lines of work we expect to work on.

In the end of the last chapter, we proposed an adaptive algorithm for quantum amplitude estimation using *approximately exact*[1] Bayesian inference, and lightly touched on its pros and cons. One of our end-goals would be to improve upon this algorithm.

For the simulation results shown herein, we used brute force greedy optimization of the experimental controls. Note that in our case, these were discrete; brute force optimization amounted to evaluating the merit of consecutive Grover exponents up to some upper bound. It is to be expected that better results could be obtained via more sophisticated strategies. Even simple heuristics to reduce the search range could drastically reduce the classical processing cost. As an example, has been shown that the optimal experimental controls in this type of problem have an exponential dependence on the iteration number [15]. This information could be a starting point for cutting back on the number of expected utility evaluations while barely affecting the results. As an example, the control could be searched for within a moving window.

Alternatively, more intricate approaches could be taken to choose experiments in a systematic fashion. One option would be to forgo the greedy strategy, opting instead for approximate global optimization with little to no online processing involved. Another would be to create a "black box" that, given the measurement results, quickly outputs a proposed control for the following iteration. This tool could be fine-tuned using training data.

Furthermore, we could attempt to broaden the scope of achievable likelihood functions by generalizing the Grover operator as in [59]. Extending the achievable likelihood functions to a continuum would open the door to otherwise ill-suited optimization methods.

In addition to this, alternative methods could be attempted for the discretization of the Bayesian distributions, numerical or otherwise. Examples include rejection filtering, variational approximations,

[1] This apparent oxymoron means to convey the fact that the sole source of inaccuracy is the finitude of the allocated resources, which stands in contrast with the inescapable limitations of e.g. variational inference.

other variants of sequential Monte Carlo, and Markov Chain Monte Carlo. The precision of the numerical or analytical representation of the distributions ultimately determines the accuracy of the estimates they produce. Hence, the quality of the representation plays a crucial role in the inference, both in the result extraction and (relatedly) in the experiment selection. This is bound to become particularly relevant as the models' complexity increases, which is inevitable if they are to absorb noise through nuisance parameters.

This takes us to another salient point, which is the adaptation of the strategy to noisy quantum systems. As just mentioned, this could mean tailoring the model; but it could also mean modifying the experiment control selection criteria, or refining the processing strategy. The ability to incorporate such revisions comes from the flexibility of the Bayesian framework. The simplest alteration would be to impose an upper cap on the circuit length implied by the chosen control. Despite its simplicity, this is expected have a strong positive impact on estimation under the presence of decoherence. But naturally, more sophisticated improvements can be had, namely through the adoption of a noise model - which too can be calibrated using experimental data.

Another direction we would like to explore are the characteristics of QAE with the application to Monte Carlo integration in mind. Some topics are unique to this application, such as the loading of probability distributions, integrand-dependent considerations, arithmetic operations, and others. We would like to study the challenges associated with this particular end-goal.

To summarize, the main tasks we set out to realize are:

- To improve robustness and/or efficiency of adaptive Bayesian techniques, by using improved discretization methods and different utility maximization strategies, preferably noise-aware and error-lenient ones.
- To attempt alternative techniques with little to no online processing, based on algorithms for approximate global optimization (such as particle swarm optimization, differential evolution, genetic algorithms, neural networks, or others).
- To further evaluate the impact of noise on different QAE algorithms, using both noise models and real devices.
- To seek and test noise resilient algorithms, and assess how much (if any) of the quantum advantage can be preserved when using NISQ devices.
- To study aspects particular to QAE as applied to Monte Carlo integration, in terms of both their general relevance and their implications for near-term devices.
- To implement quantum algorithms for Monte Carlo integration on actual quantum devices and/or classical simulators with noise models, applying them to small scale problems as close as possible to practical applications.

4.2 Work plan

Having in mind the goals outlined in the previous section, we have constructed a provisional work plan for this project, which is presented in figure 30 along with the projected timing of the tasks.

No.	Task	Year 1												Year 2												Year 3												Year 4											
		Month																																															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
1	Survey of the state of the art on quantum amplitude estimation (QAE) algorithms, including fault-tolerant and nearer-term approaches. Overview of related topics employing analogous techniques. Study of numerical integration as an application for QAE. *(Completed)*	■	■	■	■	■	■																																										
2	Development and testing of benchmarking strategies for estimation algorithms. Implementation of pre-existing QAE algorithms. Analysis of their performance via simulations. Study of their noise resilience from theoretical and numerical perspectives. Experimental assessment of the viability of implementing them in present-day quantum devices. *(Ongoing)*			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■						
3	Implementation of a Bayesian amplitude estimation algorithm. Attempt at a a locally optimal strategy via introduction of adaptivity. Analysis of advantages and disadvantages as compared to pre-existing proposals. Identification of possible improvements. Thesis proposal write-up. *(Completed)*							■	■	■																																							
4	Continued development of the algorithm. Experiment with approximate optimization strategies. Search for ways of increasing robustness while minimizing the overhead involved. Investigation of the trade-offs incurred.										■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■						
5	Theoretical and practical work for the development of a noise-resilient algorithm for QAE. Demonstration of results in practical applications, using classical data (e.g. time series of relevant processes) as well as quantum data (e.g. quantum simulation outputs, quantum sensor read-out).										■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■						
6	Inspection of topics specific to QAE-enhanced Monte Carlo integration, such as loading probability distributions, integrand types, and arithmetic operations.																									■	■	■	■	■	■	■	■	■															
7	Writing the dissertation.																																											■	■	■	■	■	■

Figure 30: Tentative timeline for the project.

Bibliography

[1] S. Aaronson and P. Rall. *Quantum Approximate Counting, Simplified*. 2020-01. doi: 10.1137/1.9781611976014.5. arXiv: 1908.10846 [quant-ph]. (Visited on 2022-10-03) (cit. on pp. 7, 16, 29, 37, 53).

[2] Abbas, Amira et al. “Learn Quantum Computation Using Qiskit”. https://community.qiskit.org/textbook/preface.htm (Visited on 2023-03-17) (cit. on p. 72).

[3] D. S. Abrams and C. P. Williams. “Fast Quantum Algorithms for Numerical Integrals and Stochastic Processes”. 1999-08. arXiv: arXiv:quant-ph/9908083. (Visited on 2022-11-22) (cit. on p. 8).

[4] G. Brassard et al. “Quantum Amplitude Amplification and Estimation”. In: vol. 305. 2002, pp. 53–74. doi: 10.1090/conm/305/05215. arXiv: quant-ph/0005055. (Visited on 2022-10-03) (cit. on pp. 1–5, 7, 8, 10, 15, 17–19, 24, 29, 30, 37, 50, 51, 73).

[5] R. E. Caflisch. “Monte Carlo and Quasi-Monte Carlo Methods”. In: *Acta Numerica* 7 (1998-01), pp. 1–49. issn: 0962-4929. doi: 10.1017/S0962492900002804. (Visited on 2022-12-06) (cit. on pp. 2, 42).

[6] K. Craigie et al. “Resource-Efficient Adaptive Bayesian Tracking of Magnetic Fields with a Quantum Sensor”. In: *Journal of Physics: Condensed Matter* 33.19 (2021-04), p. 195801. issn: 0953-8984. doi: 10.1088/1361-648X/abe34f. (Visited on 2023-01-27) (cit. on p. 57).

[7] A. Dalal and A. Katabarwa. “Noise Tailoring for Robust Amplitude Estimation”. 2022-08. arXiv: arXiv:2208.11797. (Visited on 2023-02-17) (cit. on pp. 8, 61).

[8] R. Daviet. “Inference with Hamiltonian Sequential Monte Carlo Simulators”. 2018-12. arXiv: arXiv:1812.07978. (Visited on 2023-01-27) (cit. on p. 59).

[9] P. Del Moral and A. Doucet. “Sequential Monte Carlo Samplers”. 2002-12. arXiv: arXiv:cond-mat/0212648. (Visited on 2023-01-27) (cit. on p. 59).

[10] A. Doucet et al. “Sequential Monte Carlo Methods in Practice”. In: (2013-01). issn: 978-1-4419-2887-0. doi: 10.1007/978-1-4757-3437-9_24 (cit. on p. 59).

[11] D. J. Egger et al. “Quantum Computing for Finance: State of the Art and Future Prospects”. In: *IEEE Transactions on Quantum Engineering* 1 (2020), pp. 1–24. issn: 2689-1808. doi: 10.1109/TQE.2020.3030314. arXiv: 2006.14510 [quant-ph, q-fin]. (Visited on 2022-12-06) (cit. on pp. 2, 9).

[12] J. Etxezarreta Martinez et al. “Time-Varying Quantum Channel Models for Superconducting Qubits”. In: *npj Quantum Information* 7.1 (2021-07), pp. 1–10. issn: 2056-6387. doi: 10.1038/s41534-021-00448-5. (Visited on 2023-02-17) (cit. on p. 54).

[13] E. Farhi, J. Goldstone, and S. Gutmann. “A Quantum Approximate Optimization Algorithm”. 2014-11. doi: 10.48550/arXiv.1411.4028. arXiv: arXiv:1411.4028. (Visited on 2023-01-20) (cit. on p. 4).

[14] C. Ferrie, C. E. Granade, and D. G. Cory. “Adaptive Hamiltonian Estimation Using Bayesian Experimental Design”. In: *AIP Conference Proceedings* 1443.1 (2012-05), pp. 165–173. issn: 0094-243X. doi: 10.1063/1.3703632. (Visited on 2023-01-27) (cit. on p. 59).

[15] 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”. In: *Quantum Information Processing* 12.1 (2013-01), pp. 611–623. issn: 1570-0755, 1573-1332. doi: 10.1007/s11128-012-0407-6. arXiv: 1110.3067 [quant-ph]. (Visited on 2023-01-28) (cit. on p. 62).

[16] C. Figgatt et al. “Complete 3-Qubit Grover Search on a Programmable Quantum Computer”. In: *Nature Communications* 8.1 (2017-12), p. 1918. issn: 2041-1723. doi: 10.1038/s41467-017-01904-7. (Visited on 2023-03-09) (cit. on p. 3).

[17] S. Fukuzawa et al. “Modified Iterative Quantum Amplitude Estimation Is Asymptotically Optimal”. 2022-11. arXiv: arXiv:2208.14612. (Visited on 2023-01-17) (cit. on pp. 7, 36, 37).

[18] K. Georgopoulos, C. Emary, and P. Zuliani. “Modelling and Simulating the Noisy Behaviour of Near-term Quantum Computers”. In: *Physical Review A* 104.6 (2021-12), p. 062432. issn: 2469-9926, 2469-9934. doi: 10.1103/PhysRevA.104.062432. arXiv: 2101.02109 [quant-ph]. (Visited on 2023-02-17) (cit. on p. 54).

[19] T. Giurgica-Tiron et al. “Low Depth Algorithms for Quantum Amplitude Estimation”. In: *Quantum* 6 (2022-06), p. 745. doi: 10.22331/q-2022-06-27-745. (Visited on 2023-03-20) (cit. on pp. 7, 28).

[20] A. Gómez et al. “A Survey on Quantum Computational Finance for Derivatives Pricing and VaR”. In: *Archives of Computational Methods in Engineering* 29.6 (2022-10), pp. 4137–4163. issn: 1886-1784. doi: 10.1007/s11831-022-09732-9. (Visited on 2022-12-06) (cit. on p. 2).

[21] C. E. Granade et al. “Robust Online Hamiltonian Learning”. In: *New Journal of Physics* 14.10 (2012-10), p. 103013. issn: 1367-2630. doi: 10.1088/1367-2630/14/10/103013. (Visited on 2022-10-03) (cit. on pp. 55, 59).

[22] D. Grinko et al. “Iterative Quantum Amplitude Estimation”. In: *npj Quantum Information* 7.1 (2021-12), p. 52. issn: 2056-6387. doi: 10.1038/s41534-021-00379-1. arXiv: 1912.05559 [quant-ph]. (Visited on 2022-10-03) (cit. on pp. 7, 8, 23, 24, 34, 35, 37, 50, 54, 58).

[23] L. K. Grover. “A Fast Quantum Mechanical Algorithm for Database Search”. 1996-11. doi: 10.48550/arXiv.quant-ph/9605043. arXiv: arXiv:quant-ph/9605043. (Visited on 2022-12-12) (cit. on pp. 1, 3, 4).

[24] S. Herbert. “Quantum Computing for Data Centric Engineering and Science”. In: *Data-Centric Engineering* 3 (2022), e36. issn: 2632-6736. doi: 10.1017/dce.2022.36. arXiv: 2212.02133 [quant-ph]. (Visited on 2022-12-06) (cit. on p. 42).

[25] S. Herbert. “The Problem with Grover-Rudolph State Preparation for Quantum Monte-Carlo”. In: *Physical Review E* 103.6 (2021-06), p. 063302. issn: 2470-0045, 2470-0053. doi: 10.1103/PhysRevE.103.063302. arXiv: 2101.02240 [quant-ph]. (Visited on 2023-03-19) (cit. on pp. 9, 42).

[26] D. Herman et al. “A Survey of Quantum Computing for Finance”. In: *Papers* 2201.02773 (2022-06). (Visited on 2023-03-19) (cit. on p. 9).

[27] B. L. Higgins et al. “Entanglement-Free Heisenberg-limited Phase Estimation”. In: *Nature* 450.7168 (2007-11), pp. 393–396. issn: 0028-0836, 1476-4687. doi: 10.1038/nature06257. arXiv: 0709.2996 [quant-ph]. (Visited on 2022-10-03) (cit. on p. 57).

[28] F. Huszár and N. M. T. Houlsby. “Adaptive Bayesian Quantum Tomography”. In: *Physical Review A* 85.5 (2012-05), p. 052120. doi: 10.1103/PhysRevA.85.052120. (Visited on 2023-01-27) (cit. on p. 57).

[29] P. D. Johnson et al. “Reducing the Cost of Energy Estimation in the Variational Quantum Eigensolver Algorithm with Robust Amplitude Estimation”. 2022-03. arXiv: arXiv:2203.07275. (Visited on 2022-12-07) (cit. on pp. 3, 8, 61).

[30] I. Kassal et al. “Polynomial-Time Quantum Algorithm for the Simulation of Chemical Dynamics”. In: *Proceedings of the National Academy of Sciences* 105.48 (2008-12), pp. 18681–18686. issn: 0027-8424, 1091-6490. doi: 10.1073/pnas.0808245105. arXiv: 0801.2986 [quant-ph]. (Visited on 2022-12-06) (cit. on p. 3).

[31] A. Katabarwa et al. “Reducing Runtime and Error in VQE Using Deeper and Noisier Quantum Circuits”. 2021-10. arXiv: arXiv:2110.10664. (Visited on 2022-12-07) (cit. on pp. 3, 8, 61).

[32] I. Kerenidis et al. “Q-Means: A Quantum Algorithm for Unsupervised Machine Learning”. 2018-12. doi: 10.48550/arXiv.1812.03584. arXiv: arXiv:1812.03584. (Visited on 2022-12-06) (cit. on p. 3).

[33] A. Y. Kitaev. “Quantum Measurements and the Abelian Stabilizer Problem”. 1995-11. doi: 10.48550/arXiv.quant-ph/9511026. arXiv: arXiv:quant-ph/9511026. (Visited on 2022-12-12) (cit. on p. 5).

[34] E. Knill, G. Ortiz, and R. D. Somma. “Optimal Quantum Measurements of Expectation Values of Observables”. In: *Physical Review A* 75.1 (2007-01), p. 012328. doi: 10.1103/PhysRevA.75.012328. (Visited on 2022-12-06) (cit. on p. 3).

[35] D. E. Koh et al. “Foundations for Bayesian Inference with Engineered Likelihood Functions for Robust Amplitude Estimation”. In: *Journal of Mathematical Physics* 63.5 (2022-05), p. 052202. issn: 0022-2488, 1089-7658. doi: 10.1063/5.0042433. arXiv: 2006.09349 [math-ph, physics:quant-ph]. (Visited on 2022-12-07) (cit. on pp. 8, 61).

[36] J. Liu and M. West. “Combined Parameter and State Estimation in Simulation-Based Filtering”. In: *Sequential Monte Carlo Methods in Practice*. Ed. by A. Doucet, N. de Freitas, and N. Gordon. Statistics for Engineering and Information Science. New York, NY: Springer, 2001, pp. 197–223. isbn: 978-1-4757-3437-9. doi: 10.1007/978-1-4757-3437-9_10. (Visited on 2023-01-27) (cit. on p. 59).

[37] A. Lumino et al. “Experimental Phase Estimation Enhanced By Machine Learning”. In: *Physical Review Applied* 10.4 (2018-10), p. 044033. issn: 2331-7019. doi: 10.1103/PhysRevApplied.10.044033. arXiv: 1712.07570 [quant-ph]. (Visited on 2022-10-19) (cit. on p. 73).

[38] Malcom Levitt. *Spin Dynamics: Basics of Nuclear Magnetic Resonance*. Second. Wiley, 2008. (Visited on 2023-02-06) (cit. on p. 10).

[39] R. Merlin. “Rabi Oscillations, Floquet States, Fermi’s Golden Rule, and All That: Insights from an Exactly Solvable Two-Level Model”. In: *American Journal of Physics* 89.1 (2021-01), pp. 26–34. issn: 0002-9505. doi: 10.1119/10.0001897. (Visited on 2023-02-06) (cit. on p. 10).

[40] K. Miyamoto. “Bermudan Option Pricing by Quantum Amplitude Estimation and Chebyshev Interpolation”. In: *EPJ Quantum Technology* 9.1 (2022-12), pp. 1–27. issn: 2196-0763. doi: 10.1140/epjqt/s40507-022-00124-3. (Visited on 2022-12-06) (cit. on p. 2).

[41] M. Mosca and C. Zalka. “Exact Quantum Fourier Transforms and Discrete Logarithm Algorithms”. 2003-01. arXiv: arXiv:quant-ph/0301093. (Visited on 2023-01-23) (cit. on p. 15).

[42] K. Nakaji. “Faster Amplitude Estimation”. In: *Quantum Information and Computation* 20.13&14 (2020-11), pp. 1109–1123. issn: 15337146, 15337146. doi: 10.26421/QIC20.13-14-2. arXiv: 2003.02417 [quant-ph]. (Visited on 2022-10-03) (cit. on pp. 7, 36–38, 54).

[43] M. A. Nielsen and I. L. Chuang. “Quantum Computation and Quantum Information: 10th Anniversary Edition”. https://www.cambridge.org/highereducation/books/quantum-computation-and-quantum-information/01E10196D0A682A6AEFFEA52D53BE9AE. 2010-12. doi: 10.1017/CBO9780511976667. (Visited on 2023-02-17) (cit. on p. 54).

[44] "Quantum Monte Carlo Methods in Physics and Chemistry". In: ed. by M. P. Nightingale and C. J. Umrigar. Dordrecht: Springer Netherlands, 1999. isbn: 978-0-7923-5552-6 978-94-011-4792-7. doi: 10.1007/978-94-011-4792-7. (Visited on 2023-01-24) (cit. on p. 11).

[45] A. Peruzzo et al. "A Variational Eigenvalue Solver on a Quantum Processor". In: *Nature Communications* 5.1 (2014-07), p. 4213. issn: 2041-1723. doi: 10.1038/ncomms5213. arXiv: 1304.3061 [physics, physics:quant-ph]. (Visited on 2023-01-20) (cit. on p. 4).

[46] L. Pezze' and A. Smerzi. "Quantum Theory of Phase Estimation". 2014-11. arXiv: arXiv:1411.5164. (Visited on 2023-01-26) (cit. on p. 11).

[47] K. Plekhanov et al. "Variational Quantum Amplitude Estimation". In: *Quantum* 6 (2022-03), p. 670. issn: 2521-327X. doi: 10.22331/q-2022-03-17-670. arXiv: 2109.03687 [quant-ph]. (Visited on 2022-10-19) (cit. on pp. 7, 27, 37).

[48] J. Preskill. "Quantum Computing in the NISQ Era and Beyond". In: *Quantum* 2 (2018-08), p. 79. issn: 2521-327X. doi: 10.22331/q-2018-08-06-79. arXiv: 1801.00862 [cond-mat, physics:quant-ph]. (Visited on 2023-01-20) (cit. on p. 3).

[49] N. F. Ramsey. "A Molecular Beam Resonance Method with Separated Oscillating Fields". In: *Physical Review* 78.6 (1950-06), pp. 695–699. doi: 10.1103/PhysRev.78.695. (Visited on 2023-02-06) (cit. on p. 10).

[50] P. Rebentrost, B. Gupt, and T. R. Bromley. "Quantum Computational Finance: Monte Carlo Pricing of Financial Derivatives". In: *Physical Review A* 98.2 (2018-08), p. 022321. issn: 2469-9926, 2469-9934. doi: 10.1103/PhysRevA.98.022321. arXiv: 1805.00109 [quant-ph]. (Visited on 2022-12-06) (cit. on p. 2).

[51] R. Santagati et al. *Magnetic-Field-Learning Using a Single Electronic Spin in Diamond with One-Photon-Readout at Room Temperature*. 2018-07 (cit. on pp. 11, 55).

[52] P. W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Journal on Computing* 26.5 (1997-10), pp. 1484–1509. issn: 0097-5397, 1095-7111. doi: 10.1137/S0097539795293172. arXiv: quant-ph/9508027. (Visited on 2022-12-06) (cit. on pp. 3, 4, 72).

[53] U. Skosana and M. Tame. "Demonstration of Shor's Factoring Algorithm for N $$=$$ 21 on IBM Quantum Processors". In: *Scientific Reports* 11.1 (2021-08), p. 16599. issn: 2045-2322. doi: 10.1038/s41598-021-95973-w. (Visited on 2023-03-09) (cit. on p. 3).

[54] L. F. South, A. N. Pettitt, and C. C. Drovandi. "Sequential Monte Carlo Samplers with Independent Markov Chain Monte Carlo Proposals". In: *Bayesian Analysis* 14.3 (2019-09), pp. 753–776. issn: 1936-0975, 1931-6690. doi: 10.1214/18-BA1129. (Visited on 2023-01-27) (cit. on p. 59).

[55] N. Stamatopoulos et al. “Option Pricing Using Quantum Computers”. In: *Quantum* 4 (2020-07), p. 291. issn: 2521-327X. doi: 10.22331/q-2020-07-06-291. arXiv: 1905.02666 [quant-ph]. (Visited on 2022-12-06) (cit. on p. 2).

[56] Y. Suzuki et al. “Amplitude Estimation without Phase Estimation”. In: *Quantum Information Processing* 19.2 (2020-02), p. 75. issn: 1570-0755, 1573-1332. doi: 10.1007/s11128-019-2565-2. arXiv: 1904.10246 [quant-ph]. (Visited on 2022-10-03) (cit. on pp. 7, 9, 25, 26, 28, 36, 37, 40, 51, 52, 57).

[57] K. M. Svore, M. B. Hastings, and M. Freedman. “Faster Phase Estimation”. 2013-04. arXiv: arXiv:1304.0741. (Visited on 2023-03-17) (cit. on pp. 72, 73).

[58] J. M. Taylor et al. “High-Sensitivity Diamond Magnetometer with Nanoscale Resolution”. In: *Nature Physics* 4.10 (2008-10), pp. 810–816. issn: 1745-2473, 1745-2481. doi: 10.1038/nphys1075. arXiv: 0805.1367 [cond-mat]. (Visited on 2023-01-19) (cit. on p. 11).

[59] G. Wang et al. “Minimizing Estimation Runtime on Noisy Quantum Computers”. In: *PRX Quantum* 2.1 (2021-03), p. 010346. issn: 2691-3399. doi: 10.1103/PRXQuantum.2.010346. arXiv: 2006.09350 [quant-ph]. (Visited on 2022-12-07) (cit. on pp. 8, 60–62).

[60] J. Wang et al. “Experimental Quantum Hamiltonian Learning”. In: *Nature Physics* 13.6 (2017-06), pp. 551–555. issn: 1745-2481. doi: 10.1038/nphys4074. (Visited on 2023-01-27) (cit. on p. 57).

[61] C. R. Wie. “Simpler Quantum Counting”. In: *Quantum Information and Computation* 19.11&12 (2019). issn: 15337146, 15337146. doi: 10.26421/QIC19.11-12. arXiv: 1907.08119 [quant-ph]. (Visited on 2022-12-14) (cit. on pp. 7, 33, 37, 53, 57).

[62] N. Wiebe and C. E. Granade. “Efficient Bayesian Phase Estimation”. In: *Physical Review Letters* 117.1 (2016-06), p. 010503. issn: 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.117.010503. arXiv: 1508.00869 [quant-ph]. (Visited on 2022-10-03) (cit. on p. 73).

[63] N. Wiebe, A. Kapoor, and K. Svore. “Quantum Algorithms for Nearest-Neighbor Methods for Supervised and Unsupervised Learning”. 2014-07. doi: 10.48550/arXiv.1401.2142. arXiv: arXiv:1401.2142. (Visited on 2022-12-06) (cit. on p. 3).

[64] N. Wiebe, A. Kapoor, and K. M. Svore. “Quantum Deep Learning”. 2015-05. doi: 10.48550/arXiv.1412.3489. arXiv: arXiv:1412.3489. (Visited on 2022-12-06) (cit. on p. 3).

[65] S. Wiedemann et al. “Quantum Policy Iteration via Amplitude Estimation and Grover Search – Towards Quantum Advantage for Reinforcement Learning”. 2022-06. doi: 10.48550/arXiv.2206.04741. arXiv: arXiv:2206.04741. (Visited on 2022-12-06) (cit. on p. 3).

[66] S. Woerner and D. J. Egger. “Quantum Risk Analysis”. In: *npj Quantum Information* 5.1 (2019-12), p. 15. issn: 2056-6387. doi: 10.1038/s41534-019-0130-6. arXiv: 1806.06893 [quant-ph]. (Visited on 2022-12-06) (cit. on p. 2).

[67] C. Zalka. “Grover’s Quantum Searching Algorithm Is Optimal”. In: *Physical Review A* 60.4 (1999-10), pp. 2746–2751. issn: 1050-2947, 1094-1622. doi: 10.1103/PhysRevA.60.2746. arXiv: quant-ph/9711070. (Visited on 2023-03-09) (cit. on p. 1).

A

Quantum phase estimation and Hadamard tests

Quantum phase estimation (QPE) is a key routine in quantum computing. It underpins many quantum algorithms - notably, this includes Shor's factoring algorithm [52].

Its goal is to estimate the phase ϕ of the eigenvalue λ of a unitary $\hat{\mathbf{U}}$'s eigenstate $|\ \psi\ \rangle$. The two last elements are given.

$$\hat{\mathbf{U}}\,|\ \psi\ \rangle = \lambda\,|\ \psi\ \rangle \tag{A.1}$$

$$\lambda = e^{i\phi} \tag{A.2}$$

In its most traditional form, it consists of a Hadamard transform, followed by a ladder of $\hat{\mathbf{U}}$ controlled operators, followed by a QFT. This sequence of operations is rather onerous for near-term quantum-devices. Adding to this is the reliance of the algorithm on perfect measurements, which makes it extremely susceptible to the faults that swarm experimental practice. These factors combine to produce unusable results for any application of practical interest in the near future.

An alternative approach is *iterative* phase estimation [57, 2]. While this epithet can describe many approaches, it is generally taken to mean a still-exact adaptive algorithm based on the circuit depicted in figure 31, which it embeds in a classical feedback loop.

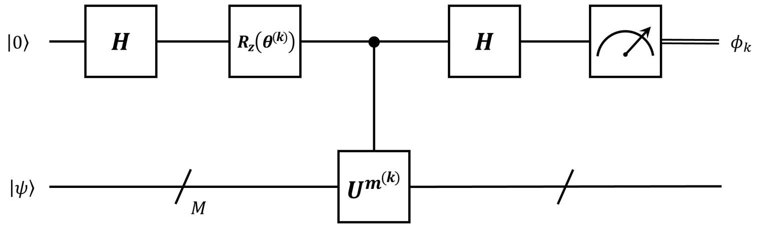


Figure 31: Iterative phase estimation circuit diagram (with k an iteration label).

The $\theta^{(k)}$ and $m^{(k)}$ variables are controllable parameters to be chosen according to the iteration. The sequence of operator powers $m^{(k)}$ is pre-determined, whereas the correction angles $\theta^{(k)}$ must be selected adaptively based on results from previous iterations. Such an arrangement allows for sequentially inferring the digits of the phase written as a binary fraction, at a rate of one per iteration and in reverse order (from least to most significant).

Importantly, this approach reduces both depth and width as compared to standard QPE. For a proper choice of parameters ($m^{(k)} = 1, \theta^{(k)} = 0$), the circuit in figure 31 matches the traditional phase estimation circuit for *one* mere auxiliary qubit (for the QFT register), whose execution would generally be a pointless exercise.

Alternative algorithms based on the same circuit have been proposed, using classical processing tools such as statistical inference or machine learning to improve noise-resilience [57, 62, 37]. Since QPE is an integral part of the original QAE algorithm, it can benefit directly from such modifications, which have much the same flavor as the near-term approaches to QAE described in the main text.

However, this does not mean that the problem of QAE boils down to QPE, or that QAE-specific approaches are superfluous. This is due to the fact that in the particular case of QAE, the state $|\, \psi \,\rangle$ is not an eigenstate of the targeted operator $\hat{\mathbf{U}}$ - in this case, the Grover operator. In spite of this, the unknown of interest - the amplitude - can still be determined from a phase measurement, due to its collapsing the state into one of the operator eigenstates, whose eigenphases ultimately produce the same result [4].

This special set of circumstances is particular to QAE, and allows QPE to work despite the fact that the operator does not act trivially on the state. To act trivially would be to impart a global phase. Instead, the Grover operator has an interesting effect, which is to amplify the amplitude; and powers thereof have a still more interesting effect, causing the amplitude to oscillate with the exponent.

Because this exponent amounts to a powerful degree of freedom, it can make for highly informative measurements, even without controlled versions of the operator (as are necessary in QPE). Hence, nearly all simplified approaches to QAE use even simpler circuits than that of figure 31, which gives them the upper hand over general QPE algorithms.

Finally, we note that for the particular case $m^{(k)} = 1, \theta^{(k)} = 0$, the circuit in 31 also corresponds to the *Hadamard test* circuit, which is used to estimate the real part of the expectation value of an operator $\hat{\mathbf{U}}$ in a state $|\, \psi \,\rangle$. The expected value of the output of the circuit in figure 31 is precisely $\mathrm{Re}\,\langle\, \psi \,|\, \hat{\mathbf{U}} \,|\, \psi \,\rangle$.