Quantum Information Processing (2024) 23:82
https://doi.org/10.1007/s11128-024-04273-5

# Functional quantum abstract detecting systems

Guillermo Lugilde[1] · Elías F. Combarro[2] · Ignacio F. Rúa[1]

Received: 2 April 2023 / Accepted: 15 January 2024 / Published online: 28 February 2024

## Abstract

Quantum abstract detecting systems (QADS) provide a common framework to address detection problems in quantum computers. A particular QADS family, that of combinatorial QADS, has been proved to be useful for decision problems on eigenvalues or phase estimation methods. In this paper, we consider functional QADS, which not only have interesting theoretical properties (intrinsic detection ability, relation to the QFT), but also yield improved decision and phase estimation methods, as compared to combinatorial QADS. A first insight into the comparison with other phase estimation methods also shows promising results.

**Keywords** Quantum abstract detecting systems · Combinatorial QADS · Functional QADS · Geometric QADS · Quantum Fourier transformation · Quantum phase estimation

## 1 Introduction

One of the tasks where quantum computers outperform classical computers is the search of marked elements in unsorted lists, thanks to Grover's algorithm [1]. Quantum abstract detecting systems (QADS) [2] provide a general quantum computing framework to address detection problems, and so they generalize Grover and various other quantum algorithms (Deutsch–Jozsa algorithm [3], the quantum abstract search [4] and several quantum walks [5–8]). QADS also allow us to combine them in order to improve their accuracy. In addition, this paradigm has potential applications when-

✉ Guillermo Lugilde
lugildeguillermo@uniovi.es

Elías F. Combarro
efernandezca@uniovi.es

Ignacio F. Rúa
rua@uniovi.es

[1] Mathematics Department, University of Oviedo, Oviedo, Spain

[2] Computer Science Department, University of Oviedo, Oviedo, Spain

ever the existence of a marked element must be found, for example, checking the commutativity of finite dimensional algebras [9]. A similar approach was explored for the detection of undesired measurements in a circuit [10].

The family of combinatorial QADS [11], which control one prefixed QADS with some extra qubits, is particularly interesting. They can not only be used in the original detecting setting, but also in some other practical problems, such as a decision algorithm for an eigenvalue of a unitary matrix or, most importantly, phase estimation by the Hadamard test. Namely, the latter approximates an eigenvalue of a given unitary matrix. A well-known method to do this is the quantum phase estimation algorithm [12], but the Hadamard test achieves it in a simpler way but with less precision, and it can be generalized with combinatorial QADS.

In this work, we introduce functional QADS, which extend the family of combinatorial QADS. Theoretically, they have some interesting properties. First, they have a $\delta$-detecting time if it is based on a QADS that has one. Also, the QFT operation can be reinterpreted as a product of Hadamard gates and geometric QADS (a particular type of functional QADS).

On the other hand, we will consider the previous combinatorial QADS-based algorithms for functional QADS, aiming to improve the probabilities of error or the efficiency. We will find that combinatorial QADS are outperformed in both aspects. Optimality of functional QADS will be shown for the decision algorithm (the rest of the methods are based on it). In addition, a whole new phase estimation algorithm is going to be presented, based on a specific type of functional QADS, which will show a much more efficient and accurate performance than the generalized Hadamard test. Its circuit is similar to that of the quantum phase estimation (QPE) one, but avoiding the use of the QFT, which has an expensive implementation.

In summary, the practical problems that are going to be addressed are:

1. Decision on eigenvalues: given a unitary matrix $U$, an eigenvector $|\varphi_0\rangle$ with eigenvalue $e^{i\beta}$ and an angle $\alpha$, decide whether $\beta = \alpha$. The study of this problem will establish the theoretical basis for the following problems.
2. Decision on an interval: given a unitary matrix $U$, an eigenvector $|\varphi_0\rangle$ with eigenvalue $e^{i\beta}$ and an interval $[\alpha - \delta, \alpha + \delta]$, decide whether $\beta \in [\alpha - \delta, \alpha + \delta]$.
3. Phase estimation problem with a confidence interval: given a unitary matrix $U$ and an eigenvector $|\varphi_0\rangle$ with eigenvalue $e^{i\beta}$, find a confidence interval for $\beta$. In particular, the following algorithms will be developed and compared:
    - Dichotomy search (for a given error in the estimation)
    - Generalized Hadamard test (for a given level of confidence; only for combinatorial QADS)
    - $\delta$-approximation algorithm (for a given error in the estimation)

After the introduction of the $\delta$-approximation algorithm, we will show some considerations about it that illustrate its potential. This includes comparing it with the QPE and other similar phase estimation methods, showing a promising performance in terms of reducing the number of qubits and operations. In addition, it offers the possibility of improving an estimation given by any other method. Also, we address how an error in the preparation of the initial state evolves in the decision algorithm.

This paper is divided in 6 sections. Section 2 contains the basic definitions and results used in our study. Section 3 will introduce $m$-functional QADS and their basic properties, paying special attention to geometric QADS. The practical applications and description of the proposed phase estimation method will be addressed in Sect. 4. Some first insights into the comparison with other phase estimation methods and the propagation of errors will be explored Sect. 5. Finally, a summary of the conclusions will be given in Sect. 6.

## 2 Preliminaries

We will summarize the basic results about QADS needed in our study of functional QADS. A QADS is a procedure meant to detect the existence of marked elements in a given set. This is achieved by an operator that fixes an initial state when the element is marked. Grover's algorithm is an example of QADS.

Hence, we define a QADS $\mathcal{Q}$ as any (classical deterministic) algorithm that takes, from a set of inputs $\mathcal{M}$, a boolean function (given by a circuit) $f : \{0, 1\}^k \longrightarrow \{0, 1\}$ and outputs a unitary transformation $U = U_f$ on a Hilbert space $\mathcal{H}$ whose dimension only depends on $k$, together with a state $|\varphi_0\rangle \in \mathcal{H}$ (that only depends on k too) such that

$$\{x \in \{0, 1\}^k \mid f(x) = 1\} = \emptyset \;\Rightarrow\; U_f|\varphi_0\rangle = |\varphi_0\rangle.$$

From this definition, a detection scheme was introduced in [2, Main algorithm] for deciding whether the received function $f$ is different from 0 or not, that is, if there exists a marked element in a given set. Namely, the initial state $|\varphi_0\rangle$ and the detecting operator $U$ from $\mathcal{Q}$ on the input $f$ are precomputed; later, $t$ is uniformly chosen from $\{0, 1, \ldots, T\}$, for a fixed value $T$, and $U^t|\varphi_0\rangle$ is measured on an orthonormal basis containing $|\varphi_0\rangle$. If the result is $|\varphi_0\rangle$, the decision $f \equiv 0$ is made; otherwise $f \not\equiv 0$.

The performance of a QADS in this algorithm is studied through the concept of $\delta$-detecting time, that characterizes the probability of error of the detecting scheme when $f \not\equiv 0$ (when $f \equiv 0$, the detection scheme never fails). If $(|\varphi_0\rangle, U = U_f)$ denotes the output of a QADS on input $f \in \mathcal{M}$, then for a given $0 < \delta \leq 1$, a function $T : \mathbb{N} \longrightarrow \mathbb{N}$ is a $\delta$-(quantum) detecting time for the QADS, if for all nonzero $f \in \mathcal{M}$ of input size $k$

$$\frac{\sum\limits_{t=0}^{T(k)} \left|\langle\varphi_0|U^t|\varphi_0\rangle\right|^2}{T(k) + 1} \leq 1 - \delta.$$

**Theorem** [2, Main theorem] *The detection scheme of the main algorithm always provides a correct output on input zero (i.e., when no marked elements do exist), and so the probability of error is fully attributed to nonzero inputs. Namely, such a probability is equal to*

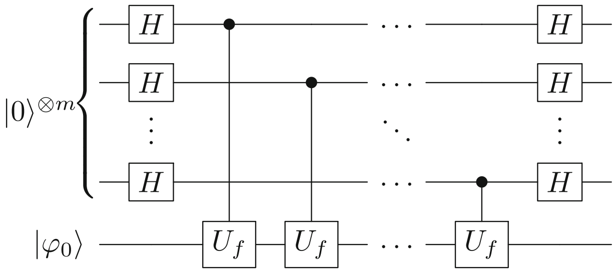


**Fig. 1** Combinatorial QADS circuit

$$\frac{\sum_{t=0}^{T(k)} \left|\langle\varphi_0|U^t|\varphi_0\rangle\right|^2}{T(k)+1}.$$

$\delta$-detecting times allow to bound the probability of error of the detection scheme. For example, Grover's algorithm has a $\frac{\sqrt{2}-1}{4\sqrt{2}}$-detecting time, of order $O(\sqrt{2^k})$.

Subsequently, $m$-combinatorial QADS were introduced in [11], which are based on a fixed QADS, adding $m$ extra qubits to it in order to control the application of its detecting operator $U_f$. If $|\varphi_0\rangle$ is its initial state and $m$ is a positive integer, then the $m$-combinatorial QADS obtained from $\mathcal{Q}$ is the QADS whose initial state is $|0\rangle^{\otimes m}|\varphi_0\rangle$, and whose detecting operator is given by

$$C(m, U_f) := (H^{\otimes m} \otimes I)c_m U_f ... c_1 U_f (H^{\otimes m} \otimes I),$$

where $c_i U_f$ is the operator $U_f$ controlled by the $i$-th qubit of the first register. Its circuit is shown in Fig. 1.

The final state of the circuit is given in the next result.

**Proposition** [11, Proposition 2] *The amplitude of the state $C(m, U_f)|0\rangle^{\otimes m}|\varphi_0\rangle$ related to the basis state $|0\rangle^{\otimes m}|\varphi_0\rangle$ is*

$$\frac{1}{2^m}\sum_{k=0}^{m}\binom{m}{k}\langle\varphi_0|U^k|\varphi_0\rangle.$$

Apart from the intended detecting nature of QADS, combinatorial QADS have other applications, among them (see [11]):

*Decision on eigenvalues* Let $U$ be a unitary matrix, and let $|\varphi_0\rangle$ be a state under the promise that $U|\varphi_0\rangle = e^{i\beta}|\varphi_0\rangle$. Then, for a given $\alpha$, we want to decide whether $\alpha = \beta$. The problem can be solved by implementing $C(m, V)$, where $V = e^{-i\alpha}U$. If the result of a final measure is $|0\rangle^{\otimes m}|\varphi_0\rangle$ (the initial state), we conclude $\beta = \alpha$. Otherwise, we conclude $\beta \neq \alpha$.

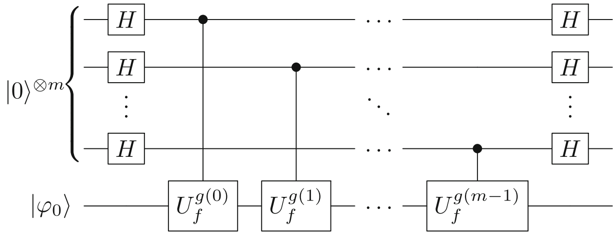


**Fig. 2** Functional QADS circuit

**Theorem** [11, Theorem 3] *The decision on eigenvalues algorithm is always correct when it outputs NO. So, the probability of error is fully attributed to a YES answer. Namely, such a probability is equal to*

$$\cos^{2m}\left(\frac{\beta-\alpha}{2}\right).$$

In this paper, we will study the behaviour of this algorithm for functional QADS, and find the optimal ones.

*Dichotomy search.* With the previous notation, a small interval containing $\beta \in [0, \pi]$ is to be found. The interval $[0, \pi]$ is split into two halves, finding the one half where $\beta$ is more likely to be. The repetition of this process as many times as desired yields the desired small interval.

In this paper, we will take a closer look to this algorithm under the paradigm of functional QADS.

*m-Hadamard test* It is the generalization of the Hadamard test, a phase estimation algorithm that approximates the angle $\beta \in [0, \pi]$, under the previous notation. It takes advantage from the fact that the formula $\cos^{2m}\left(\frac{\beta}{2}\right)$ is easily invertible. So, $n$ independent measurements of the final state of the $m-$combinatorial QADS gives $\beta \approx \arccos(2\sqrt[m]{\hat{p}_n}-1)$, where $\hat{p}_n$ is the proportion of $|0\rangle^{\otimes m}|\varphi_0\rangle$ states obtained from the decision algorithms. It has been concluded that $m = 1$ provides the most balanced version of this method for an unknown $\beta$, in the sense that increasing $m$ would improve the accuracy when $\beta \approx 0$ but make it worse when $\beta \approx \pi$.

In this paper, these last two algorithms will be considered from functional QADS in order to deal with phase estimation.

## 3 *m*-Functional QADS definition and first results

In this section, we provide the formal definition of $m$-functional QADS, which extends the idea of combinatorial QADS introduced in [11]. The $m$-functional QADS are built from another QADS, by a control of the original detecting operator $U_f$ by a superposition of qubits and by a function $g$ as shown in Fig. 2.

**Definition 1** If $U_f$ is the detecting operator of a QADS $\mathcal{Q}$, $|\varphi_0\rangle$ its initial state, $m$ is a positive integer and $g : \mathbb{N} \longrightarrow \mathbb{Q}$ a function (where $0 \in \mathbb{N}$), we define the

$m$-functional QADS obtained from $\mathcal{Q}$ as the QADS with initial state $|0\rangle^{\otimes m}|\varphi_0\rangle$ and detecting operator

$$F(m, U_f, g) := (H^{\otimes m} \otimes I)c_m U_f^{g(m-1)} ... c_1 U_f^{g(0)} (H^{\otimes m} \otimes I),$$

where $c_i U_f$ is the unitary operator that applies $U_f$ to the second register if the $i$-th qubit of the first register is $|1\rangle$, and applies the identity if that qubit is $|0\rangle$ (i.e., it is the operator $U_f$ controlled by the $i$-th qubit of the first register). The size of an $m$-functional QADS is defined as the number of times that $U_f$ is applied, that is, $G = \sum_{n=0}^{m-1} g(n)$.

We let $g$ output rational numbers in order to allow powers and roots of QADS. The $m$-combinatorial QADS are a particular case of $m$-functional QADS when $g(n) = 1$, so $C(m, U_f) = F(m, U_f, 1)$.

Before getting into the application of functional QADS considered in this paper, which is phase estimation, it is reasonable to study the basic properties of functional QADS, as well as their performance for the original QADS purpose. We first confirm that $m$-functional QADS are indeed QADS and, then, we also consider under which algorithmic operations, of those in the algorithmic closure of a QADS, (collected in Table 1, as taken from [2]), the familiy of $m-$functional QADS is closed. Proofs of these facts, and several others along the paper can be found in 1.

**Proposition 1** *Every m-functional QADS is indeed a QADS.*

**Proposition 2**
- *Extension, inversion, powers and roots of m-functional QADS are also m-functional QADS.*
- *The product of m-functional QADS built from the same original QADS is also an m-functional QADS.*
- *The product of m-functional QADS built from different original QADS is also an m-functional QADS, as long as they share the initial state and the function g, and the detecting operators involved commute with each other.*

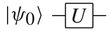
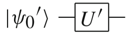
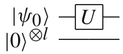
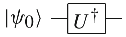
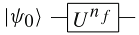
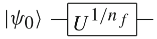
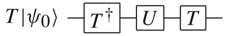
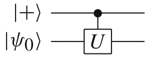
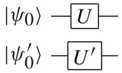
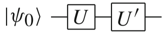
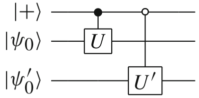
The next result provides the amplitude of the initial state at the end of the circuit. This will be a key element in order to calculate the probability of error of the detection algorithm, and of the practical applications developed later in the paper. Observe that when $g = 1$, we recover the result for $m$-combinatorial QADS given in [11].

**Theorem 1** *Given an m-functional QADS, the amplitude of the state $F(m, U_f, g)$ $|0\rangle^{\otimes m}|\varphi_0\rangle$ associated to the basis state $|0\rangle^{\otimes m}|\varphi_0\rangle$ is*

$$\frac{1}{2^m} \sum_{x=0}^{2^m-1} \langle\varphi_0|U_f^B|\varphi_0\rangle,$$

*where $B = \sum_{i=0}^{m-1} x_i g(i)$ and $x = x_0 + 2x_1 + ... + 2^{m-1}x_{m-1}$ with $x_i \in \{0, 1\}, \forall i = 0, ..., m-1$.*

**Table 1** Transformations in the algorithmic closure of a QADS

| Name | Initial state | Detecting operator | Circuit |
|---|---|---|---|
| QADS #1 | $\|\psi_0\rangle$ | $U$ |  |
| QADS #2 | $\|\psi_0{}'\rangle$ | $U'$ |  |
| Extension | $\|\psi_0\rangle\|0\rangle^{\otimes l}$ | $U \otimes I$ |  |
| Inversion | $\|\psi_0\rangle$ | $U^\dagger$ |  |
| Powers | $\|\psi_0\rangle$ | $U^{nf}$ |  |
| Roots | $\|\psi_0\rangle$ | $U^{1/nf}$ |  |
| Conjugation | $T\|\psi_0\rangle$ | $TUT^\dagger$ |  |
| Controlled | $\|+\rangle\|\psi_0\rangle$ | $U_c\|i\rangle\|x\rangle = \|i\rangle U^i\|x\rangle$ |  |
| Tensor product | $\|\psi_0\rangle\|\psi_0{}'\rangle$ | $U \otimes U'$ |  |
| Product | $\|\psi_0\rangle(= \|\psi_0'\rangle)$ | $U'U$ |  |
| Doubly controlled | $\|+\rangle\|\psi_0\rangle\|\psi_0'\rangle$ | $U_{dc}\|i\rangle\|x\rangle\|x'\rangle = \|i\rangle U^i\|x\rangle U'^{1-i}\|x'\rangle$ |  |

One of the situations in which this amplitude is needed is when determining $\delta$-detecting functions. For a QADS, the existence of a $\delta$-detecting function, implies the ability of bounding the probability of error of their main detection algorithm. Thus, it is a key feature for a QADS to be used in detection problems providing an error bound. As stated by the following result, under certain conditions, if a fixed QADS has a $\delta_T$-detecting time, then any $m$-functional QADS has its own $\delta_S$-detecting time.

**Theorem 2** *Let $m > 0$, $M = 2^m$ and $G = \sum_{i=0}^{m-1} g(i)$ for a given function $g$. If $T : \mathbb{N} \to \mathbb{N}$ is a $\delta_T$-detecting time such that $1 - \delta_T < 1/M$, and $T(k) > M - 1$, then any $m$-functional QADS of size $G \leq \frac{MT(k)}{T(k)+1-M}$ will have $S(k) = \left\lfloor \frac{T(k)}{G} \right\rfloor$ as a $\delta_S$-detecting time, where $\delta_S = 1 - M(1 - \delta_T)$.*

It is worth pointing out that $\frac{MT(k)}{T(k)+1-M} \geq \frac{MT(k)}{T(k)} = 2^m$, so in order to guarantee the existence of a $\delta-$detecting time, it is enough to ensure that its size is at most $2^m$.

We are going to focus on three types of functional QADS, which we will introduce now, mainly because of their behaviour on the decision algorithm. Recall that the performance of the practical algorithms later developed in the paper are strongly based on it.

*Combinatorial QADS*: $g_c(n) = 1$

If we consider the function $g_c$ to be constant and equal to 1, then we obtain the already studied $m$-combinatorial QADS. They are the simplest and easiest to work with, and they provide an analytically invertible probability of a positive outcome (which provides the approximation of the eigenvalue's angle $\beta$, as stated above).

*Linear QADS*: $g_l(n) = n + 1$

$m-$functional QADS with $g_l(n) = n + 1$ will be called $m$-linear QADS. They have a predictable behaviour and are usually a better option over the combinatorial QADS for certain problems. It is the best functional QADS among those whose probability of a positive outcome for the decision algorithm is always decreasing with respect to $|\beta - \alpha|$ in $[0, \pi]$. This is a desirable property in several situations.

*Geometric QADS*: $g_g(n) = 2^n$

$m-$geometric QADS are $m-$functional QADS with $g_g(n) = 2^n$, which have proven to be the best choice for all the studied applications, as its performance on the decision algorithm is optimal. We will also prove that the Quantum Fourier Transformation can be nearly completely explained in terms of a product of geometric QADS.

### 3.1 Geometric QADS

We will dedicate this subsection to geometric QADS (i.e., $g_g(n) = 2^n$), due to their overall superior performance for the practical applications seen in the next section. Figure 3 shows its circuit.

From the perspective of implementation, it is worth pointing out that, when $U_f$ is a rotation, the corresponding power $U_f^{2^i}$ is also a rotation (with different angle), so the geometric QADS does not require the application of an exponential number of gates. If $U_f$ is not a rotation, then, for most practical applications, the fact that the number

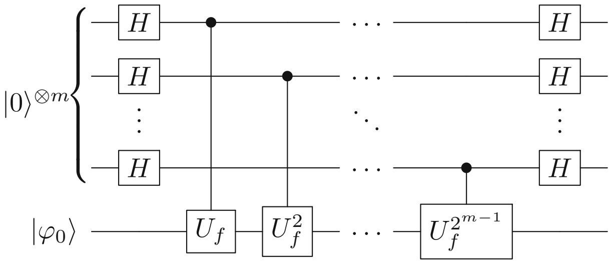


**Fig. 3** Geometric QADS circuit

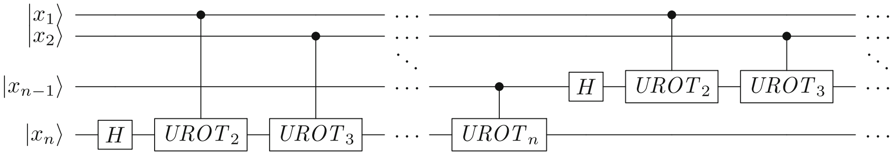


**Fig. 4** QFT circuit

of qubits needed to obtain an accuracy $\delta$ increases logarithmically on $\delta$ counteracts the exponential increasing of applications of $U_f$.

For geometric QADS, Theorem 1 can be directly rewritten in the following way.

**Corollary 1** *In the case of an m-geometric QADS, the amplitude of the state $F(m, U_f, g_g)|0\rangle^{\otimes m}|\varphi_0\rangle$ related to the basis state $|0\rangle^{\otimes m}|\varphi_0\rangle$ is*

$$\frac{1}{2^m}\sum_{x=0}^{2^m-1}\langle\varphi_0|U_f^x|\varphi_0\rangle.$$

This formula will be especially useful in the computation of later probabilities of error, since it yields expressions involving a geometric sum (hence the name given to the QADS).

In addition, let us show that the quantum Fourier transformation circuit can be described by the composition of a series of geometric QADS. If we denote

$$UROT_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{pmatrix},$$

then the circuit implementing $QFT|x_1x_2...x_n\rangle$ is like the one in Fig. 4. Because $UROT_k^2 = UROT_{k-1}$ and, consequently, $UROT_k^{2^p} = UROT_{k-p}$, we can describe the QFT circuit in the way shown in Fig. 5.

Observe the appearance of a sequence of partial geometric QADS (with $m$ decreasing) plus an initial application of a $H^{\otimes n}$ gate. This is because the $UROT_k$ gates commute with each other. Hence, this proves that

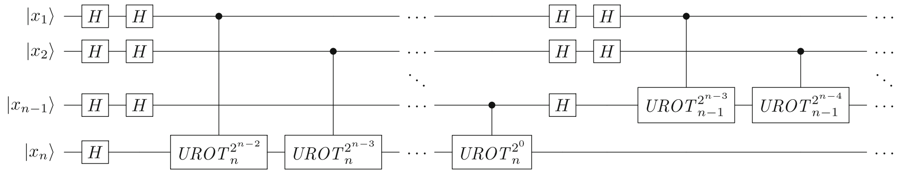


**Fig. 5** QFT circuit as a sequence of geometric QADS

$$QFT_n = \left(F(0, UROT_1, g_g) \otimes I^{\otimes n-1}\right)\left(F(1, UROT_2, g_g) \otimes I^{\otimes n-2}\right) \\ \dots \left(F(n-1, UROT_n, g_g)\right) H^{\otimes n}. \tag{1}$$

In addition, since the inversion of a geometric QADS is also a geometric QADS, we can obtain an analogous formula for $QFT_n^\dagger$. This connection illustrates the generality of QADS, and could be helpful for including the QFT gate in the framework and, therefore, in the circuits. This way, algorithms such as the QPE [12, page 224] could be studied through a new perspective and compared to the phase estimation method introduced in the following section.

## 4 Functional QADS: practical applications

In this section, we consider functional QADS in the context of the phase estimation problem. We aim at an approximation of the phase of an eigenvalue of a unitary matrix $U$, along with a confidence interval of desired length. Our proposed algorithm solving this problem is based on a chain of several simpler algorithms that will be detailed first. We will begin with an algorithm for a decision problem on eigenvalues and, based on it, we construct a decision algorithm on intervals. An improvement to the latter leads us to the final phase estimation method: the $\delta$-approximation algorithm.

### 4.1 Decision on eigenvalues

In this problem, we are given a unitary matrix $U$, a state $|\varphi_0\rangle$, under the promise that $U|\varphi_0\rangle = e^{i\beta}|\varphi_0\rangle$ for an unknown real number $\beta$. Then, for a given $\alpha \in \mathbb{R}$, we shall use a functional QADS-based decision algorithm (DA) that checks whether $\beta = \alpha$. Since all of the following algorithms are based on this one, it is worth studying it deeply. In this setting, we will see that geometric QADS are the best choice. The algorithm is the following.

**Algorithm 1** Decision on eigenvalues for an $m$-functional QADS (DA)

```
Given an $m$-functional QADS, a state $|\varphi_0\rangle$, a unitary matrix $U$ such that $U|\varphi_0\rangle = e^{i\beta}|\varphi_0\rangle$ and an angle $\alpha$:
Implement $F(m, V, g)$ based on the matrix $V = e^{-i\alpha}U$, and apply it to the initial state $|0\rangle^{\otimes m}|\varphi_0\rangle$.
Measure the final state on any orthonormal basis containing $|0\rangle^{\otimes m}|\varphi_0\rangle$.
If the result is $|0\rangle^{\otimes m}|\varphi_0\rangle$, output ‘YES’. If the result is any other state different from $|0\rangle^{\otimes m}|\varphi_0\rangle$, output ‘NO’.
```

Based on this procedure, we can prove a formula for the probability of stating that $\alpha = \beta$. The proofs of the results of this section can be found in 2.

**Theorem 3** *Under the promise that $U|\varphi_0\rangle = e^{i\beta}|\varphi_0\rangle$, given an angle $\alpha$ and any $m$-functional QADS for $U$, the probability of a positive outcome from the decision algorithm is*

$$DA(m, g, \beta - \alpha) := \prod_{n=0}^{m-1} \cos^2\left(g(n)\frac{\beta - \alpha}{2}\right).$$

This theorem, when $g = 1$, yields the analogous result in the case of combinatorial QADS [11, Theorem 3]. Moreover, the following particular result for geometric QADS can be obtained too.

**Theorem 4** *Under the promise that $U|\varphi_0\rangle = e^{i\beta}|\varphi_0\rangle$, given an angle $\alpha$ and an $m$-geometric QADS for $U$, the probability of a positive outcome from the decision algorithm when $\beta \neq \alpha$ is*

$$DA(m, g_g, \beta - \alpha) = \frac{1 - \cos(2^m(\beta - \alpha))}{2^{2m}(1 - \cos(\beta - \alpha))}.$$

As a consequence, we have:

**Corollary 2** *In the previous conditions, $DA(m, g_g, \frac{2k\pi}{2^m}) = 0$, for any $k \in \{1, ..., 2^m - 1\}$.*

The probabilities of stating that $\alpha = \beta$ of the DA for a fixed $m = 5$ can be seen in Fig. 6. We observe how both the linear and the geometric clearly outperform the combinatorial QADS, being the geometric the best choice, especially when $\beta$ and $\alpha$ are close to each other. However, it could be argued that both the linear and the geometric QADS have greater sizes than the combinatorial one. So, despite their better probabilities, the corresponding circuits are not as efficient. To address this, we study the performance of the different functional QADS when a size $G$ is fixed, allowing $m$ to vary from one family to another.

For a fixed $m_c$, in the combinatorial QADS, $U$ is applied $m_c$ times; for a certain $m_g$, in the geometric QADS, $U$ is applied $\sum_{n=0}^{m_g-1} 2^n = \frac{1-2^{m_g}}{1-2} = 2^{m_g} - 1$ times; for a certain $m_l$, in the linear QADS, $U$ is applied $\sum_{n=0}^{m_l-1}(n+1) = \frac{m_l(m_l+1)}{2}$ times. Hence, if we fix a size $G$, then we use $U$ as much as $m_c = G$, $m_g = \log_2(G+1)$, and $m_l$ such that $m_l^2 + m_l - 2\,G \leq 0$. Obviously, some of them have to be rounded sometimes. The new graph, for a fixed size of 31 operations ($m_c = 31$, $m_g = 5$, and $m_l$ rounded to 7), can be checked in Fig. 7. Although the combinatorial and linear QADS have better performances, the geometric QADS are still best, specially for close values of $\beta$ and $\alpha$. Moreover, they use significantly less qubits.

However, the probability is not always lower. We shall prove that, for a fixed $m$ and in average, the probability of a positive outcome for the geometric QADS is always smaller than for any other functional QADS. To do this, it should be noticed that,

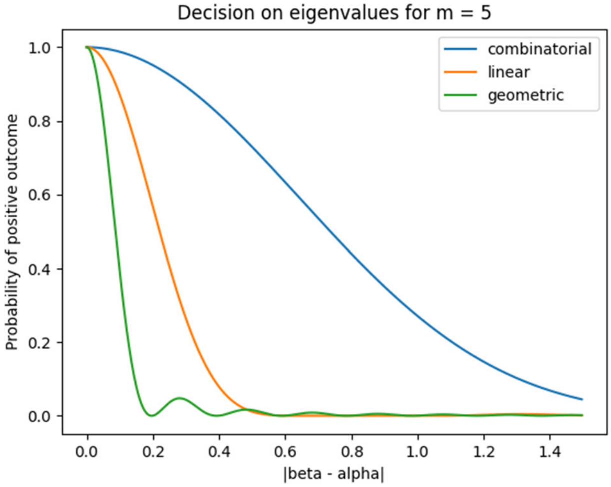


**Fig. 6** Probability of a positive outcome from the decision algorithm for the combinatorial, linear and geometric QADS and a fixed $m = 5$

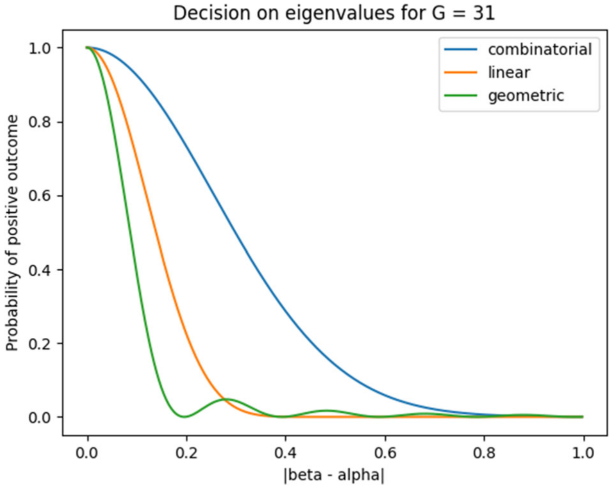


**Fig. 7** Probability of a positive outcome from the decision algorithm for the combinatorial, linear and geometric QADS and a fixed size of 31 operations

in the DA setting, only functional QADS with a natural valued function $g$ are worth considering. Negative numbers can be ruled out, since the cosines of $\mathrm{DA}(m, g, \beta - \alpha)$ are not affected by a change of sign. Also, since it is desirable that the formula is equal to 1 when $|\beta - \alpha| = 2\pi$, we need that $g(n)$ is always an integer number. Consequently, we introduce the following definitions.

**Definition 2** We say that a natural valued function $g : \mathbb{N} \longrightarrow \mathbb{N}$ is DA-optimal for a fixed $m = m_0$ if, for any other natural valued function $h : \mathbb{N} \longrightarrow \mathbb{N}$,

$$\int_0^\pi \mathrm{DA}(m_0, g, t)\, dt \leq \int_0^\pi \mathrm{DA}(m_0, h, t)\, dt.$$

An $m_0$-functional QADS is DA-optimal for a fixed $m = m_0$, if its associated function $g : \mathbb{N} \longrightarrow \mathbb{N}$ is DA-optimal for a fixed $m = m_0$.

**Definition 3** Let $G_0 > 0$ be a natural number, $g : \mathbb{N} \longrightarrow \mathbb{N}$ a natural function such that $\sum_{n=0}^{m_g-1} g(n) = G_0$ for some $m_g > 0$. We say $g$ is DA-optimal for a fixed $G = G_0$ if for any other natural valued function $h : \mathbb{N} \longrightarrow \mathbb{N}$ such that $\sum_{n=0}^{m_h-1} h(n) = G_0$ (for some $m_h > 0$),

$$\int_0^\pi \mathrm{DA}(m_g, g, t)\, dt \leq \int_0^\pi \mathrm{DA}(m_h, h, t)\, dt.$$

An $m$-functional QADS of size $G_0$ is DA-optimal, if its associated function $g : \mathbb{N} \longrightarrow \mathbb{N}$ is DA-optimal for a fixed $G = G_0$.

Observe that the integrals are defined on the interval $[0, \pi]$, instead of $[0, 2\pi]$, because $\mathrm{DA}(m, g, t)$ is symmetrical with respect to $t = \pi$, when $g$ is natural valued. In this setting, we can prove the following result related to the optimality for a fixed $m$.

**Theorem 5** *The $m$-geometric QADS is DA-optimal for any fixed $m$. Moreover, among those $m$-functional QADS which are DA-optimal for $m$, $m$-geometric QADS have the smallest size.*

Notice that this result captures the average performance of the geometric QADS, and so there might be some functional QADS (for instance, $g(n) = (n+1)^2$), which may have a better behaviour for particular values of $|\beta - \alpha|$. However, the knowledge of zeros of $\mathrm{DA}(m, g_g, t)$ makes geometric QADS easier to handle, and so they will be used henceforth.

On the other hand, the DA-optimality for a fixed size will be studied numerically. Since there are only $m$-geometric QADS of sizes $G = 2^m - 1$, we introduce the concept of $G$-shortened geometric QADS for any other size.

**Definition 4** For a given natural $G > 0$, we define the $G$-shortened geometric QADS as an $m$-functional QADS where $m = \lceil log_2(G+1) \rceil$, $g(n) = 2^n$ when $n \neq m-1$ and $g(m-1) = G - (2^{m-1} - 1)$.

For example, the shortened geometric QADS for $G = 18$ would feature $m = 5$ and $g([0, 4]) = \{1, 2, 4, 8, 3\}$. Shortened geometric QADS are the closest functional QADS to a geometric QADS, for a given size (if $G = 2^m - 1$ for some $m$, shortened geometric QADS are $m$-geometric QADS). Many times, especially when $G \approx 2^m - 1$, they are DA-optimal for $G$ or close to optimality. For instance, shortened geometric QADS are DA-optimal for every $G$ from 1 to 19, except for $G = 12$ (in this case, optimality is achieved by a functional QADS with $g([0, 4]) = \{1, 1, 1, 3, 6\}$).

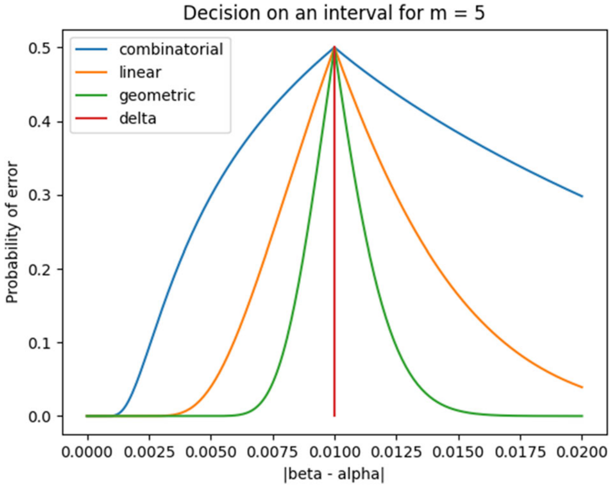


**Fig. 8** Probability of error of the decision on an interval algorithm for the combinatorial, linear and geometric QADS for a fixed $m = 5$, sample size of 1000, and $\delta = 0.01$

### 4.2 Decision on an interval

The DA can be transformed in order to check whether the eigenvalue $\beta$ belongs to a given interval $[\alpha - \delta, \alpha + \delta]$, i.e., $\alpha$ would approximate $\beta$ up to an error of $\delta$. The idea behind this algorithm is to take a decision based on $P_\delta := \mathrm{DA}(m, g, \delta)$ and the estimation of $\mathrm{DA}(m, g, \beta - \alpha)$ by a series of independent runs of the functional-based DA, $P_\alpha$. These probabilities can be estimated by the proportion of times that the QADS algorithm gave the initial state as the outcome. If the approximation $P_\alpha$ is greater than $P_\delta$, it will be assumed that the distance between $\alpha$ and $\beta$ is lower than $\delta$, and so $\beta \in [\alpha - \delta, \alpha + \delta]$. The details on the design rationale of this and the following algorithms, their actual implementations and performances, can be found in 2. Figures 8 and 9 show the performance of the studied functional QADS. Once again, the geometric QADS is the most efficient, as it inherits its behaviour from the DA.

### 4.3 Interval correction (IC)

Since the probability of error of the previous algorithm is 0.5 when $\beta$ is on one of the endpoints of the interval, an improvement can be made. Its improved version, called IC, will be the base of the $\delta$-approximation algorithm. When $P_\alpha \approx P_\delta$, we will test which endpoint $\beta$ is closer to. Suppose, for instance, that $\beta$ is close to the $\alpha + \delta$ endpoint. Then, it will be assumed that $[\alpha, \alpha + 2\delta]$ (an interval of the same length) contains $\beta$.

The details on the implementation and computation of the error probabilities can be found in 2. Figure 10 shows the behaviour of different functional QADS, giving the geometric QADS the best performance. On the other hand, the differences between the two versions of the algorithm can be seen in Figs. 11, 12 and 13. In all cases, $\delta$ has

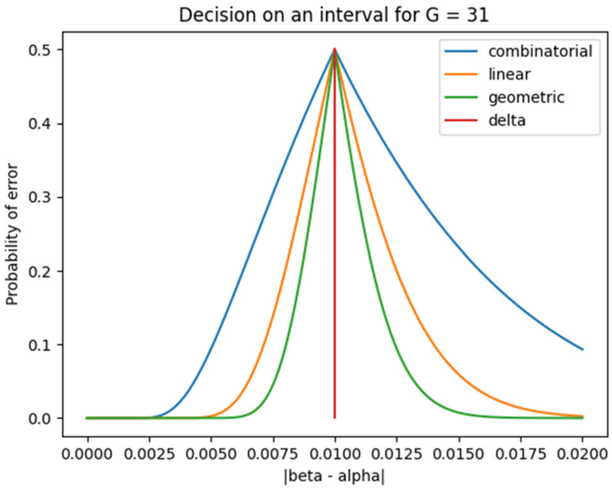


**Fig. 9** Probability of error of the decision on an interval algorithm for the combinatorial, linear and geometric QADS for a fixed $G = 31$, sample size of 1000 and $\delta = 0.01$

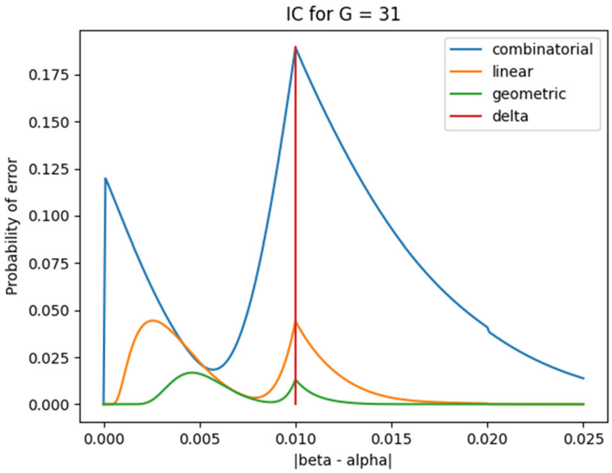


**Fig. 10** Probability of error of the IC for the combinatorial, linear and geometric QADS for a fixed $G = 31$, sample size of 1000, $\delta = 0.01$, and optimal $d_1, d_2$

been taken equal to 0.01, the sample size is 1000, the fixed number of operations is 31, and the reference probabilities have been optimally chosen. As we can observe, the improvement is remarkable overall. However, the chance of obtaining $P_\alpha \approx P_\delta$, but deciding the incorrect endpoint afterwards, might make the IC less accurate for low values of $|\beta - \alpha|$.

### 4.4 $\delta$-Approximation algorithm

The final algorithm provides an approximation of the angle $\beta$, with a certain given error $\delta$, and it works for any $\beta \in [0, 2\pi]$ (an advantage over some other versions based on

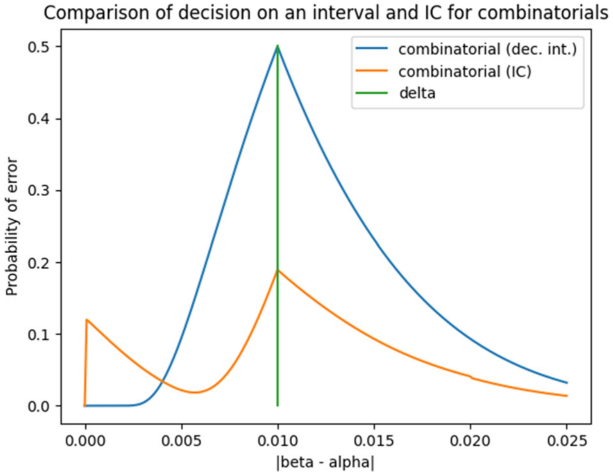


**Fig. 11** Comparison of the probability of error of the decision on an interval and IC algorithms for the combinatorial QADS of size $G = 31$, sample size of 1000, $\delta = 0.01$ and optimal $d_1, d_2$

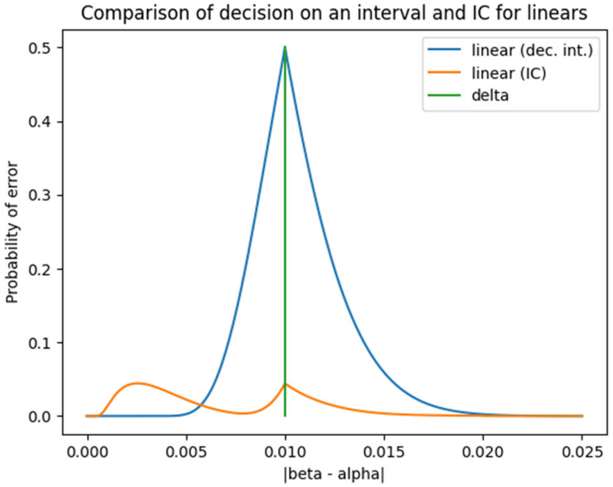


**Fig. 12** Comparison of the probability of error of the decision on an interval and IC algorithms for the linear QADS of size $G = 31$, sample size of 1000, $\delta = 0.01$, and optimal $d_1, d_2$

combinatorial QADS). The algorithm is as follows: An initial choice $\delta_0 >> \delta$ is taken, for instance, $\delta_0 = 10^2\delta$. For different approximations $\alpha_0 \in [0, 2\pi]$ of the angle $\beta$, the IC is run until the decision $\beta \in [\alpha_0 - \delta_0, \alpha_0 + \delta_0]$ is taken for a certain $\alpha_0$. In the next step of the algorithm, $\delta_1 = \frac{\delta_0}{10}, \alpha_1 = \alpha_0$ initially and we run the IC until it decides that $\beta \in [\alpha_1 - \delta_1, \alpha_1 + \delta_1]$ for a certain $\alpha_1$. In the last step of the algorithm, $\delta_2 = \frac{\delta_1}{10} = \delta$ and $\alpha_2$ is chosen analogously until the IC decides that $\beta \in [\alpha_2 - \delta_2, \alpha_2 + \delta_2]$ for a certain $\alpha_2$.

An important detail to notice is that this algorithm starts by approximating $\beta$ with low precision with the Hadamard test (which will be discussed soon, in Sect. 5.1)

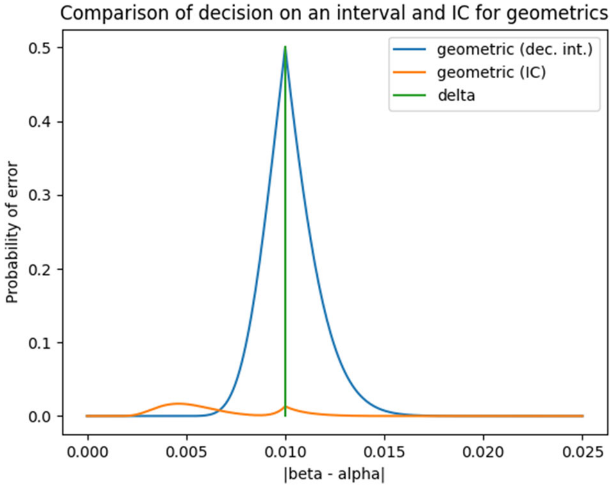


**Fig. 13** Comparison of the probability of error of the decision on an interval and IC algorithms for the geometric QADS of size $G = 31$, sample size of 1000, $\delta = 0.01$, and optimal $d_1, d_2$

before checking intervals with the IC. A more detailed account of this algorithm is given in 2. However, any other phase estimation method can be applied for that first approximation. This means that the $\delta$-approximation algorithm may offer the possibility of improving the estimation of any phase estimation method without investing a lot of qubits or measurements, as we will see in the next section.

We will stick to the Hadamard test in this article due to its direct relation to functional QADS, but more variations will be explored in the future.

With respect to the asymptotic behaviour, it was run for values of $\delta$ between $1/2^6$ and $1/2^{14}$ in order to observe how its maximum number of ancilla qubits, size and depth evolve for $p = 1/\delta$. The maximum $m$ used for $p = 2^k$ was $k + 2$, suggesting its $O(\log_2 4p)$ behaviour. The evolution of the size and depth can be observed in Fig. 14, concluding that the size evolves similarly to $O(p \log^2(\log p))$, whereas the depth (assuming that powers of $U$ are efficiently implementable) showed a similar behaviour to $O(\log^3 p)$. Particularly for the size, a clear stepped behaviour is observed, where changes of step coincide with the need of a new ancilla qubit.

However, there are some parameters in this algorithm that are yet to be optimized, so this asymptotic behaviour is just illustrative. Its theoretical behaviour will be explored in future works.

# 5 Other relevant considerations about the $\delta$-approximation algorithm

## 5.1 Comparison with other functional QADS-based methods

A natural starting point for comparing the $\delta$-approximation algorithm would be against other phase estimation methods that were previously studied for combinatorial QADS [11] which provide confidence intervals. We will introduce two of them first.

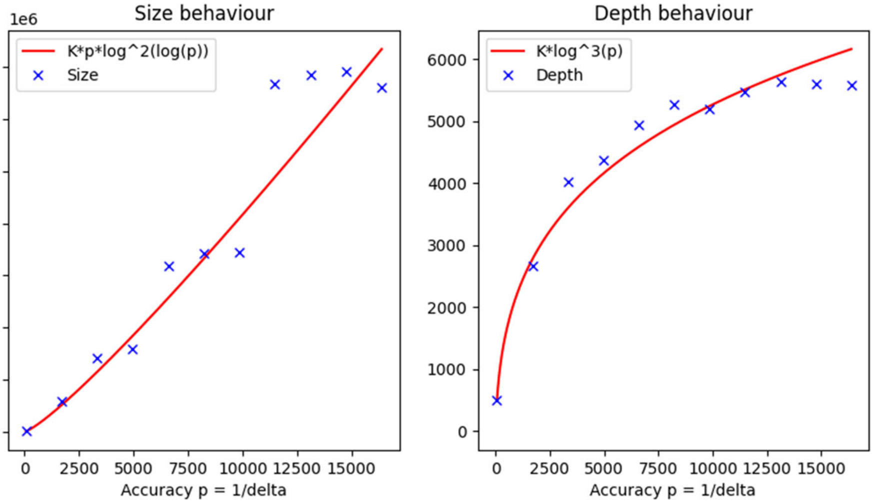


**Fig. 14** Asymptotic behaviour of the size and depth of the $\delta$-approximation method. The constants selected so that the three functions were equal for the first dot ($p = 2^6$). We also assume all repetitions of QADS to be done sequentially

*Confidence interval for the m-Hadamard test* The $m$-Hadamard test was introduced in [11]. With the same notation as above, its aim is an approximation of $\beta$. Next, we compute a confidence interval for the angle $\beta$ based on combinatorial QADS. In this test, we made $n$ independent repetitions of a combinatorial QADS experiment for the matrix $U$, and approximate $\beta$ using the fact that the probability of obtaining $|\varphi_0\rangle$ at the end of the experiment is

$$p = \cos^{2m}\frac{\beta}{2} = \left(\frac{1+\cos\beta}{2}\right)^m .$$

Each experiment follows a Bernoulli distribution, so we can apply the known expression for a confidence interval for $n$ repetitions of a Bernoulli experiment, with a preset level of confidence $a$ and an unknown standard deviation. Here, $t_{a/2}$ represents a number such that $P(T < -t_{a/2}) = a/2$, where $T$ follows a Student's $t$-distribution with $n-1$ degrees of freedom (see [13], chapter 11.4).

$$a = P\left(\overline{X} - t_{a/2}\frac{\hat{S}}{\sqrt{n}} < p < \overline{X} + t_{a/2}\frac{\hat{S}}{\sqrt{n}}\right) = P\left(\overline{X} - t_{a/2}\frac{\hat{S}}{\sqrt{n}} < \left(\frac{1+\cos\beta}{2}\right)^m\right.$$

$$\left. < \overline{X} + t_{a/2}\frac{\hat{S}}{\sqrt{n}}\right) =$$

$$= P\left(2\sqrt[m]{\overline{X} - t_{a/2}\frac{\hat{S}}{\sqrt{n}}} - 1 < \cos\beta < 2\sqrt[m]{\overline{X} + t_{a/2}\frac{\hat{S}}{\sqrt{n}}} - 1\right) =$$

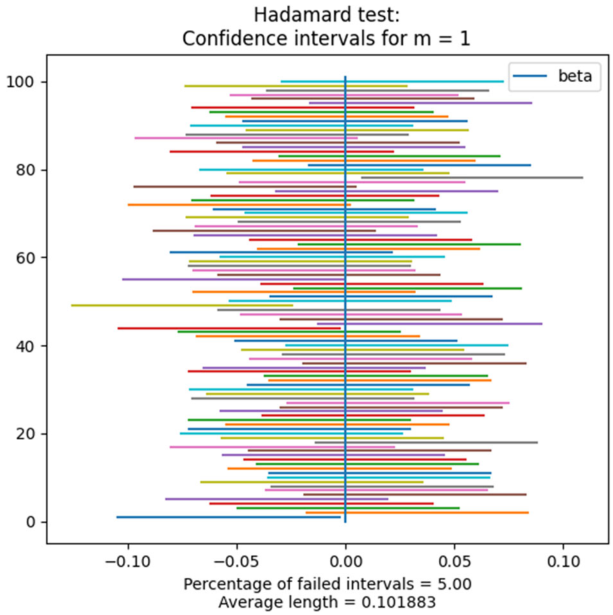


**Fig. 15** 100 confidence intervals from the Hadamard test ($m = 1$), for random $\beta$ and a sample of size 1500

$$= P\left(\arccos\left(2\sqrt[m]{\overline{X} + t_{a/2}\frac{\hat{S}}{\sqrt{n}}} - 1\right) < \beta < \arccos\left(2\sqrt[m]{\overline{X} - t_{a/2}\frac{\hat{S}}{\sqrt{n}}} - 1\right)\right)$$

Therefore, the confidence interval for $\beta$ with a level of confidence $a$ is

$$\left[\arccos\left(2\sqrt[m]{\overline{X} + t_{a/2}\frac{\hat{S}}{\sqrt{n}}} - 1\right), \arccos\left(2\sqrt[m]{\overline{X} - t_{a/2}\frac{\hat{S}}{\sqrt{n}}} - 1\right)\right]$$

The study on combinatorial QADS showed that the original Hadamard test, with $m = 1$, was the most balanced option when $\beta$ is unknown. An example of this experiment for $n = 1500$ repetitions of the QADS, $a = 0.05$, $m = 1$ and random values of $\beta$ can be seen in Fig. 15, along with the average length of the intervals. Observe that this confidence interval methodology requires that $\beta \in [0, \pi]$, although it can be adapted for greater values of $\beta$ with the approach addressed in 4.4.

*Dichotomy search* The dichotomy search assumes that $\beta$ is in the interval $[0, \pi]$, which will be split into two halves. The values of $\mathrm{DA}(m, g, \beta - \alpha)$, for $\alpha = 0$ and $\alpha = \pi$, will be approximated by a computation of multiple DA on each endpoint, and then obtaining the proportion of positive outcomes. A decision is then taken ($\beta \in [0, \frac{\pi}{2}]$ or $\beta \in [\frac{\pi}{2}, \pi]$), and the process is repeated as many times as desired. The performance of the method benefits from bigger differences in the probabilities at the endpoints. So, in order for the geometric QADS to work, we require $\mathrm{DA}(m, g_g, t)$ to be decreasing

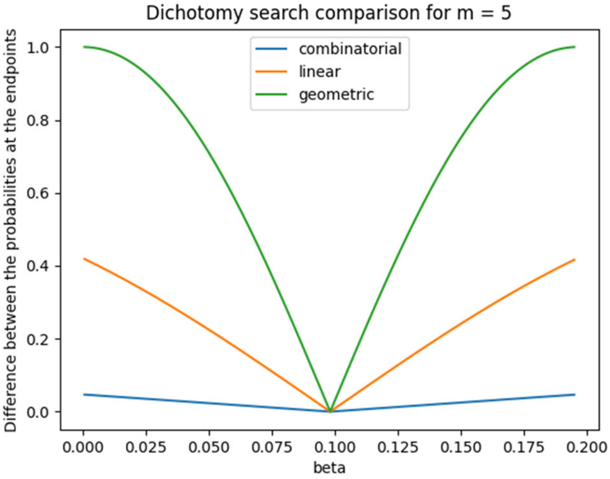


**Fig. 16** Difference in the probability of a positive outcome at the two endpoints of the interval in the fourth step of the dichotomy search for the combinatorial, linear and geometric QADS and a fixed $m = 5$

in the current interval. Therefore, for the $n$-th step of the algorithm, when the interval length is $\pi/2^n$, we should pick $m_g$ such that

$$\frac{2\pi}{2^{m_g}} \geq \frac{\pi}{2^n} \Leftrightarrow 2^{n+1} \geq 2^{m_g} \Leftrightarrow m_g \leq n+1.$$

With this choice, the performances of the three types of functional QADS under study, for the fourth step of the algorithm, and for fixed $m = 5$ and $G = 31$, are plotted in Figs. 16 and 17, respectively. As before, it can be checked that geometric QADS are the family with better performance.

*Comparing the three functional QADS-based methods.* In Fig. 18, we can see the comparison between the dichotomy search, Hadamard test and $\delta$-approximation algorithm, where the latter clearly outperforms the other two for a choice of $\delta = 0.01$ and approximately 18,000 uses of the matrix $U$. For the dichotomy search, we considered the number of iterations needed to end up with an interval of length $\approx 2\delta$; for the Hadamard test, we fixed $a = 0.05$.

## 5.2 Comparison with QPE method

Since phase estimation is the main problem addressed, it is reasonable to, at least, have a first glance at a comparison between $\delta$-approximation algorithm and the other well-known phase estimation algorithms, mainly, the QPE, whose circuit can be checked in Fig. 19.

We can observe that the circuit is exactly like the one of geometric QADS, with the exception of the final inverse of the QFT gate. We already saw in Fig. 4 how the $UROT_k$ gates are applied in the QFT. The QPE circuit with $t$ qubits would apply $U$

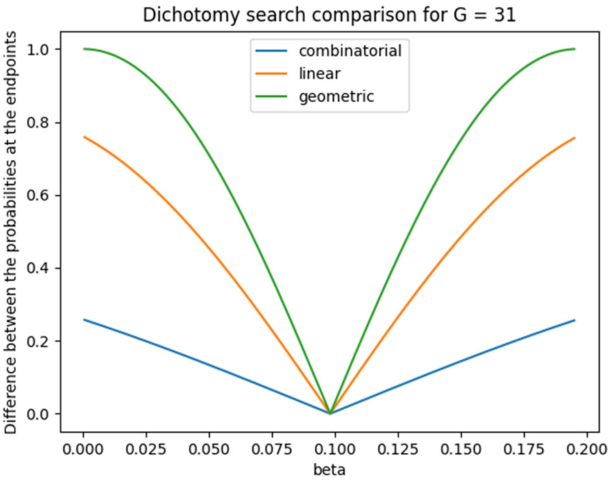


**Fig. 17** Difference in the probability of a positive outcome at the two endpoints of the interval in the fourth step of the dichotomy search for the combinatorial, linear and geometric QADS and a fixed $G = 31$

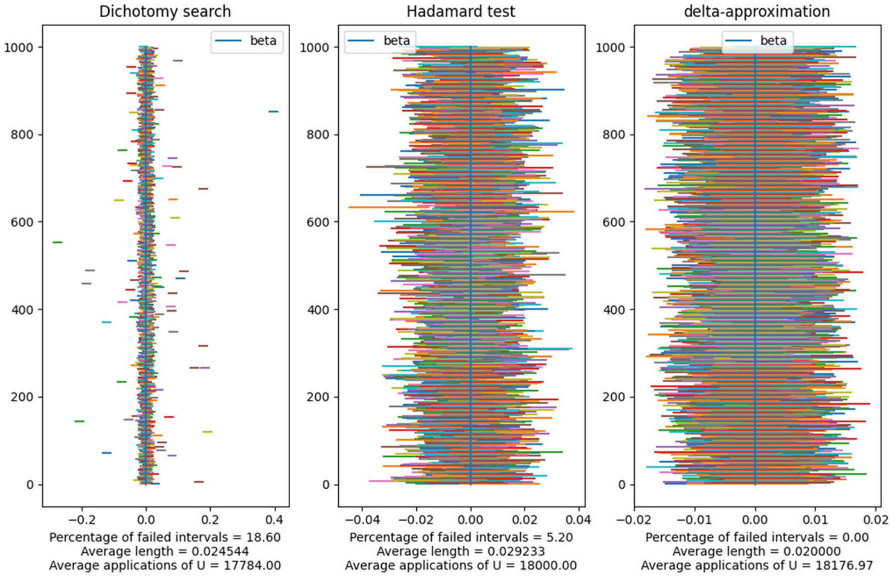


**Fig. 18** Comparison of 1000 confidence intervals of the dichotomy search, Hadamard test and $\delta$-approximation algorithm, respectively

$2^t - 1$ times, and some $UROT_k$ $(t-1)t/2$ times, $\forall k = 1, \dots, t$. That makes a total of $2^t - 1 + (t-1)t/2$.

Despite applying controlled gates ($U$ gates plus $UROT$ gates) $O(t^2)$ more times, we know that QPE estimates the phase in a single run, whereas the $\delta$-approximation algorithm needs many repetitions of geometric QADS. However, on the other hand, the accuracy of the $\delta$-approximation algorithm is higher for the same number of qubits.

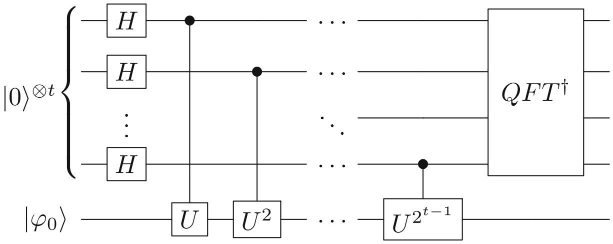


**Fig. 19** QPE circuit

Thus, a more fair comparison would take into account the number of controlled gates of both algorithms when we fix a certain accuracy. Sometimes, the exponential powers of $U$ are efficiently implementable, in the sense that there is no need to apply $U$ exponentially many times, so this comparison would address rest of the situations.

The number of qubits needed in the QPE to obtain a certain accuracy is known [12, page 224]. In order to approximate $\beta$ to an accuracy of $2^{-n}$ with a probability of success of at least $1-\epsilon$, we should take

$$t = n + \left\lceil \log_2\left(2 + \frac{1}{2\epsilon}\right) \right\rceil$$

Therefore, let us fix, for example, $n = 7$ and run 10,000 times the $\delta$-approximation algorithm for $\delta = 2^{-n} = 1/128$ to obtain a numeric approximation of its probability of error, which would represent $\epsilon$. The result is shown in Fig. 20. The error obtained is $\epsilon = 19/10000$, so

$$t = 7 + \left\lceil \log_2\left(2 + \frac{5000}{19}\right) \right\rceil = 7 + \lceil 8, 0507 \rceil = 16.$$

Without the rounding, the result of $t$ is usually around 15, so we will stick to that more generous number. Then the QPE circuit applies controlled gates $2^{15} - 1 + 14 \times 15/2 = 32782$ times, significantly higher than the average 20316.98 of the $\delta$-approximation algorithm. In addition, the latter uses half the qubits (four for the first iteration, eight for the second). Therefore, the advantage is clear when there are no efficient implementations of exponential powers of $U$.

In the case that they were efficiently implementable, still, we know that the depth of a circuit is one of the greatest concerns when avoiding errors. The longer the circuit, the more error we will get. The possibility of beating the accuracy of QPE in NISQ (noisy intermediate-scale quantum) devices, especially when using a lot of qubits, with a simpler and shorter algorithm was pointed out, for example, in [14–18]. In that sense, the idea to estimate the phase through a short circuit, even if run multiple times, can be preferable under certain situations in order to avoid these imprecisions. Also, the reduction of qubits alleviates this further.

It is true that, by the use of semiclassical Fourier transform (Fig. 21), the preparation of qubits can be delayed and the measurements can be applied earlier, using just two qubits simultaneously and reducing the time they are involved. However, besides the fact that these actions do not decrease the number of operations and can be taken in

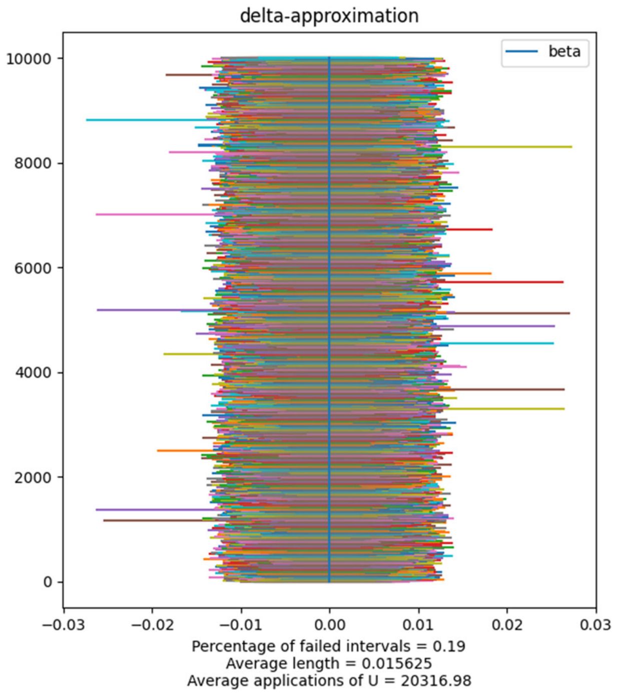


**Fig. 20** 10,000 confidence intervals of the $\delta$-approximation algorithm for $\delta = 1/128$

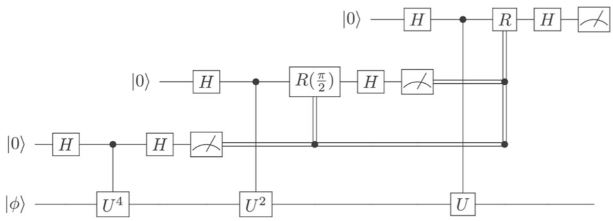


**Fig. 21** The QPE circuit with a semiclassical QTF$^\dagger$. Figure taken from [19]

the $\delta$-approximation algorithm too, there is no way of reducing the involvement of the qubits associated to $|\varphi_0\rangle$ and $U$, so the circuit length would have a negative effect anyway.

Finding a theoretical distribution for this algorithm, in order to compare it more generally and independently from numeric approximations, deserves a deeper study.

### 5.3 Comparison with other methods

*Optimal states.* An alternative approach to the QPE is introduced in [19], where the optimal states and measurement basis for phase estimation are presented, along with an error distribution. For any $N \in \mathbb{N}$, the optimal state is given by the formula:

**Table 2** Number of trials out of 100,000 with $|\beta - \alpha| \leq 1/(2^l \times 3)$. Table taken from [20]

| | Number of iterative stages, $l$ | | | |
|---|---|---|---|---|
| $N_{\text{tot}}$ | 6 | 7 | 8 | 9 |
| 20 | 99,792 | 99,729 | 99,747 | 99,712 |
| 30 | 99,993 | 99,987 | 99,982 | 99,978 |
| 40 | 99,999 | 100,000 | 99,998 | 99,999 |
| 50 | 100,000 | 100,000 | 99,999 | 100,000 |

$$|\psi_{opt}\rangle = \sqrt{\frac{2}{N+2}} \sum_{n=0}^{N} \sin\left(\frac{\pi(n+1)}{N+2}\right)|n\rangle.$$

The optimal measurement POVM (positive operator-valued measure) has elements $(N+1)/2\pi|\hat{\phi}\rangle\langle\hat{\phi}|d\phi$, where

$$|\hat{\phi}\rangle = \frac{1}{\sqrt{N+1}} \sum_{n=0}^{N} e^{in\hat{\phi}}|n\rangle,$$

and allows us to obtain an error distribution of

$$\frac{1}{\pi(N+2)}\left(\frac{\cos((\hat{\phi}-\phi)(1+N/2))\sin(\pi/(2+N))}{\cos(\pi/(2+N)) - \cos(\hat{\phi}-\phi)}\right)^2.$$

In this situation, the system would need at least $N = 1675$ to achieve an error below 1/128 with probability 19/10000. This would mean that $m > 10$, compared to the $m = 8$ of the $\delta$-approximation algorithm. Besides, the measurements have to be approximated with adaptive measurements and, as pointed out before, the depth of the circuit will cause increasing noise and, therefore, a higher chance of ruining the results.

*Iterative phase estimation* [20]. Another interesting way of estimating a phase is the iterative phase estimation, where the number of measurements needed is decreased substantially. The author provides Table 2 for presenting the results, depending on the number of iterations, $l$, and the number of measurements in each of them, $N_{tot}$.

By inputting $\delta = 1/(2^l \times 3)$ into the $\delta$-approximation algorithm, we found that, for the four different values of $l$ considered in the table, it succeeds between 99,810 and 99,880 intervals, which clearly indicates that, at least, 30 measurements per iteration would be needed for the iterative method to beat it.

Sticking to the case $l = 7$, that would mean 210 measurements, while our algorithm performs around 6155 measurements. Even though the difference is huge, 6016 of those are invested in the initial Hadamard test to approximate $\beta$, and only 139 in the rest. Still, those 139 measurements reduce the error from 1/10 to $1/(2^7 \times 3)$. This emphasizes the fact that the second part of the $\delta$-approximation algorithm shows great potential to improve a previous estimation obtained by any method.

**Table 3** Asymptotic behaviour of the Kitaev's method, faster phase estimation and $\delta$-approximation method depending on $m$m the number of powers of $U$ used

| | Width | Depth | Size |
|---|---|---|---|
| Kitaev's | $O(m\log m)$ | $O(m\log m)$ | $O(m\log m)$ |
| Faster Phase Estimation | $O(m\log^* m)$ | $O(m\log^* m)$ | $O(m\log^* m)$ |
| $\delta$-Approximation | $O(m)$ | $O(m^3)$ | $O(m^3)$ |

*Kitaev's method* [21] *and Faster Phase Estimation* [22]. Finally, we compare the $\delta$-approximation method with the Kitaev's method and Faster Phase Estimation method asymptotically. In [22, Table I] we can check the asymptotic evolution in width, depth and size of these two methods depending on $m$, the number of powers of $U$ used, assuming it is efficiently implementable and using a sequential circuit.

In the case of the $\delta$-approximation method, the width would clearly be $O(m)$. We already saw that the depth, in this case, seems similar to $O(\log^3 p)$, where $p$ is the accuracy, and evolves exponentially as $m$ increases. That would mean an $O(m^3)$ depth. The size is equivalent to the depth when $U$ is efficiently implementable.

The comparison between the three methods can be checked in Table 3. The $\delta$-approximation algorithm offers an advantage in width, and even though it seems to lose the race in terms of depth and size, we should keep in mind that, not only its theoretical asymptotic behaviour is yet unknown, but its efficiency depends on some parameters that are yet to be optimized, so the approximation in Fig. 14 is based on a particular case.

### 5.4 Inexact states

Another reasonable concern is the algorithm's behaviour when not every component is exact. How an error in the preparation of the eigenvector evolves through the DA circuit is summarized in the following result.

**Theorem 6** *In the DA for geometric QADS, suppose $|\varphi_0\rangle$ is the exact eigenvector of $U$, with eigenvalue $e^{i\beta}$, that we are interested in, but the initial inexact state $|\psi\rangle$ satisfies $|\langle\varphi_0|\psi\rangle|^2 = 1-\delta$. If $\widetilde{DA}(m, g_g, |\psi\rangle, U, \alpha)$ is the new probability of error of the DA initialized in $|\psi\rangle$, then:*

- *If $DA(m, g_g, \beta-\alpha) \le \widetilde{DA}(m, g_g, |\psi\rangle, U, \alpha)$, then*

$$\left|DA(m, g_g, \beta-\alpha) - \widetilde{DA}(m, g_g, |\psi\rangle, U, \alpha)\right| < \delta^2. \tag{2}$$

- *If $DA(m, g_g, \beta-\alpha) > \widetilde{DA}(m, g_g, |\psi\rangle, U, \alpha)$, then*

$$\left|DA(m, g_g, \beta-\alpha) - \widetilde{DA}(m, g_g, |\psi\rangle, U, \alpha)\right| < 4(\delta-\delta^2). \tag{3}$$

In 3, the theorem is proven and a further study of bound (3) is done, which shows that a more realistic bound would be $3.428\delta - 3.093\delta^2$. However, from Lemma 2 (in 3) we can see that, if $\lambda_k$ are the eigenvalues of $U$, with $\lambda_0 = \beta$, then $\forall k > 0$,

- if $\lambda_k = \beta$, then $\left|\mathrm{DA}(m, g_g, \beta - \alpha) - \widetilde{\mathrm{DA}}(m, g_g, |\psi\rangle, U, \alpha)\right| = 0$;
- if $\lambda_k = \beta \pm \pi/(2^m - 1)$, then $\left|\mathrm{DA}(m, g_g, \beta - \alpha) - \widetilde{\mathrm{DA}}(m, g_g, |\psi\rangle, U, \alpha)\right|$ might rise to $3.428\delta - 3.093\delta^2$;
- if $\lambda_k = \beta \pm \pi/2^{m-1}$, then $\left|\mathrm{DA}(m, g_g, \beta - \alpha) - \widetilde{\mathrm{DA}}(m, g_g, |\psi\rangle, U, \alpha)\right| \leq 2\delta - \delta^2$.

As we can see, the greatest error is caused when all eigenvalues fall under the second case, and, if we are aware of this situation, it can be avoided completely by just using one less qubit. Also, we are working with the difference between two functions, so even in that worst-case scenario, the bound would be approached in a very small range of values of $\alpha$, and usually for $\alpha \approx \beta$, when the algorithms have almost perfect accuracy, so the error would have a less worrying effect. Finally, since this is an error on $\mathrm{DA}(m, g_g, \beta - \alpha)$, which is a probability function, a different probability does not imply a different outcome; there is still the chance of getting the correct result anyway and, therefore, the error would have no consequences.

Nevertheless, it is obvious that a deeper study on this propagation for the IC and beyond is necessary and will be addressed in future works, as well as considering other types of errors.

## 6 Conclusions and future work

In this work, we deepen the study of the QADS framework introduced in [2], and extended in [11]. QADS, beyond their original detection purpose, have proven to be also useful in some practical applications related to phase estimation of eigenvalues of a unitary matrix.

We have introduced the new family of functional QADS, and studied basic properties, such as the amplitude of the initial state at the end of the circuit, or finding conditions for them to have a $\delta$-detecting time. In addition, the class of geometric QADS has been shown to be especially suitable for applications, and the explanation of the Quantum Fourier Transformation. For instance, geometric QADS are optimal for the decision algorithm on an eigenvalue of a unitary matrix for a fixed number of qubits and for a fixed number of $2^n - 1$ operations.

In future works, we want to explore more applications of functional QADS, and study variations of those here introduced. For instance, a change on the initial or final Hadamard gates by other rotations like QFT, which would lead to other known algorithms like the Quantum Phase Estimation.

Also, the $\delta$-approximation algorithm deserves a more thorough study in future works. There are still some possible improvements to be made, as well as optimizing the selection of certain variables that affect its accuracy and size, such as the factor that separates every $\delta_i$ from $\delta_{i+1}$ or the number of operations invested in finding $\alpha_0$. Using the algorithm as a way of improving the accuracy of any given phase estimation method is also worth studying. Although it shows competitive results when only short circuits can be executed, a theoretical distribution for this algorithm would be really helpful for comparisons with other phase estimation methods and for studying the propagation of errors in the circuit, and will be explored in future works.

**Acknowledgements** This work was partially supported by MCIN/AEI/10.13039/501100011033 under grants PID2020-119082RB-C22 and PID2021-123461NB-C22, by Gobierno del Principado de Asturias under Grant FC-GRUPIN-IDI/2021/000047 and AYUD/2021/50994, and by the Quantum Spain project funded by the Ministry of Economic Affairs and Digital Transformation of the Spanish Government and the European Union through the Recovery, Transformation and Resilience Plan - NextGenerationEU.

**Funding** Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

**Data availability** Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

## Declarations

**Conflict of interest** The authors have no competing interests to declare that are relevant to the content of this article.

## Appendix A: Proofs

### A.1.: Results in Sect. 3.

**Proposition 1** *Every $m$-functional QADS is indeed a QADS.*

***Proof*** If $f \equiv 0$, we have $U_f|\varphi_0\rangle = |\varphi_0\rangle$, so the controlled $U_f$ have no effect on the initial state. As the Hadamard gates cancel out, the initial state is always fixed when $f \equiv 0$. □

**Proposition 2**

- *Extension, inversion, powers and roots of $m$-functional QADS are also $m$-functional QADS.*
- *The product of $m$-functional QADS built from the same original QADS is also an $m$-functional QADS.*
- *The product of $m$-functional QADS built from different original QADS is also an $m$-functional QADS, as long as they share the initial state and the function $g$, and the detecting operators involved commute with each other.*

***Proof*** Extension and inversion are trivial to verify. Powers are easily verified: a functional QADS to the $k$-th power is equivalent to setting $g_p(n) = k \cdot g(n)$; analogously for roots with $g_r(n) = g(n)/k$.

In the case of the product of two $m$-functional QADS built from the same original QADS, if $g_1$ and $g_2$ are the functions of the $m$-functional QADS, the resulting $g$ function would be $g(n) = g_1(n) + g_2(n)$.

For products of $m$-functional QADS built from different original QADS, the resulting $m$-functional QADS would feature the operator $U_f V_f$, and the same function $g$. Notice that we need to make sure that both matrices commute, so that we can commute the controlled gates and transform $U_f^{g(n)} \cdot V_f^{g(n)}$ into $(U_f V_f)^{g(n)}$. □

**Theorem 1** *Given an $m$-functional QADS, the amplitude of the state $F(m, U_f, g)|0\rangle^{\otimes m}|\varphi_0\rangle$ associated to the basis state $|0\rangle^{\otimes m}|\varphi_0\rangle$ is*

$$\frac{1}{2^m}\sum_{x=0}^{2^m-1}\langle\varphi_0|U_f^B|\varphi_0\rangle,$$

*where $B = \sum_{i=0}^{m-1} x_i g(i)$ and $x = x_0 + 2x_1 + ... + 2^{m-1}x_{m-1}$ with $x_i \in \{0, 1\}, \forall i = 0, ..., m-1$.*

***Proof*** The circuit first applies $H^{\otimes m}$ to the initial state:

$$|0\rangle^{\otimes m}|\varphi_0\rangle \longrightarrow \frac{1}{\sqrt{2^m}}\sum_{x=0}^{2^m-1}|x\rangle|\varphi_0\rangle$$

Afterwards, the sequence of controlled $U_f$ are applied. In order to check the behaviour of every qubit independently, we decompose $x$ in its binary expansion: $x = x_0 + 2x_1 + ... + 2^{m-1}x_{m-1}$ with $x_i \in \{0, 1\}, \forall i = 0, ..., m-1$. For each $x$, $U_f$ is going to be applied $g(i)$ times only if $x_i = 1$. Consequently, $U_f$ is going to be applied $B = \sum_{i=0}^{m-1} x_i g(i)$ times.

$$\frac{1}{\sqrt{2^m}}\sum_{x=0}^{2^m-1}|x\rangle|\varphi_0\rangle \longrightarrow \frac{1}{\sqrt{2^m}}\sum_{x=0}^{2^m-1}|x\rangle U_f^B|\varphi_0\rangle$$

Now, the amplitude related to the basis state $|0\rangle^{\otimes m}|\varphi_0\rangle$ is:

$$\begin{aligned}&\langle\varphi_0|\langle 0|^{\otimes m}(H^{\otimes m}\otimes I)\left(\frac{1}{\sqrt{2^m}}\sum_{x=0}^{2^m-1}|x\rangle U_f^B|\varphi_0\rangle\right)\\&=\left(\frac{1}{\sqrt{2^m}}\sum_{y=0}^{2^m-1}\langle\varphi_0|\langle y|\right)\left(\frac{1}{\sqrt{2^m}}\sum_{x=0}^{2^m-1}|x\rangle U_f^B|\varphi_0\rangle\right)=\\&=\frac{1}{2^m}\sum_{x=0}^{2^m-1}\langle\varphi_0|U_f^B|\varphi_0\rangle.\end{aligned}$$

□

**Theorem 2** *Let $m > 0$, $M = 2^m$ and $G = \sum_{i=0}^{m-1} g(i)$ for a given function $g$. If $T : \mathbb{N} \to \mathbb{N}$ is a $\delta_T$-detecting time such that $1 - \delta_T < 1/M$, and $T(k) > M - 1$, then any $m$-functional QADS of size $G \leq \frac{MT(k)}{T(k)+1-M}$ will have $S(k) = \left\lfloor \frac{T(k)}{G} \right\rfloor$ as a $\delta_S$-detecting time, where $\delta_S = 1 - M(1 - \delta_T)$.*

***Proof*** If $U_c$ denotes the detecting operator of the functional QADS, and $U_f$ is the detecting operator of the QADS it is based on,

$$\frac{\sum\limits_{s=0}^{S(k)} \left|\langle\psi_0|U_c^s|\psi_0\rangle\right|^2}{S(k)+1} = \frac{\sum\limits_{s=0}^{S(k)} \left|\frac{1}{M}\sum\limits_{x=0}^{M-1}\langle\varphi_0|U_f^{sB}|\varphi_0\rangle\right|^2}{S(k)+1}.$$

The numerator can be simplified by the following inequality.

$$\left|\sum_{i=1}^{M} x_i\right|^2 = \left|\sum_{i=1}^{M} x_i\right|\left|\sum_{i=1}^{M} x_i\right| \leq \left(\sum_{i=1}^{M}|x_i|\right)\left(\sum_{i=1}^{M}|x_i|\right) = \left(\sum_{i=1}^{M}|x_i|\right)^2 \leq M\sum_{i=1}^{M}|x_i|^2$$

The last inequality follows from $0 \leq (a-b)^2 \Rightarrow 2ab \leq a^2 + b^2$.

$$\frac{\sum\limits_{s=0}^{S(k)} \left|\sum\limits_{x=0}^{M-1}\langle\varphi_0|U_f^{sB}|\varphi_0\rangle\right|^2}{M^2(S(k)+1)} \leq \frac{\sum\limits_{s=0}^{S(k)}\sum\limits_{x=0}^{M-1} \left|\langle\varphi_0|U_f^{sB}|\varphi_0\rangle\right|^2}{M(S(k)+1)} = \sum_{x=0}^{M-1}\frac{\sum\limits_{s=0}^{\lfloor T(k)/G\rfloor} \left|\langle\varphi_0|U_f^{sB}|\varphi_0\rangle\right|^2}{M\left(\left\lfloor\frac{T(k)}{G}\right\rfloor+1\right)}$$

For a fixed $x$,

$$\sum_{i=0}^{m-1} x_i g(i) \leq G \Rightarrow s\sum_{i=0}^{m-1} x_i g(i) \leq S(k)\cdot G \leq T(k),$$

and since no powers are repeated, the summation from 0 to $T(k)$ only adds positive terms to the numerator, so

$$\sum_{x=0}^{M-1}\frac{\sum\limits_{s=0}^{\lfloor T(k)/G\rfloor} \left|\langle\varphi_0|U_f^{sB}|\varphi_0\rangle\right|^2}{M\left(\left\lfloor\frac{T(k)}{G}\right\rfloor+1\right)} \leq \sum_{x=0}^{M-1}\frac{\sum\limits_{t=0}^{T(k)} \left|\langle\varphi_0|U_f^{t}|\varphi_0\rangle\right|^2}{M\left(\left\lfloor\frac{T(k)}{G}\right\rfloor+1\right)}.$$

For the denominator, it is enough if we notice that

$$M\left(\frac{T(k)}{G}+1\right) \geq T(k)+1 \Leftrightarrow \frac{T(k)}{G} \geq \frac{T(k)+1}{M} - 1 \Leftrightarrow G \leq \frac{MT(k)}{T(k)+1-M}.$$

Therefore, the hypothesis of the theorem allow us to conclude

$$\sum_{x=0}^{M-1} \frac{\sum_{t=0}^{T(k)} \left|\langle\varphi_0|U_f^t|\varphi_0\rangle\right|^2}{M\left(\left\lfloor\frac{T(k)}{G}\right\rfloor+1\right)} \leq \sum_{x=0}^{M-1} \frac{\sum_{t=0}^{T(k)} \left|\langle\varphi_0|U_f^t|\varphi_0\rangle\right|^2}{T(k)+1} \leq \sum_{x=0}^{M-1}(1-\delta_T) = M(1-\delta_T) \leq 1.$$

Finally, from $1-\delta_S = M(1-\delta_T)$, we get $\delta_S = 1 - M(1-\delta_T)$. □

### A.2.: Results in Sect. 4.

**Theorem 3** *Under the promise that $U|\varphi_0\rangle = e^{i\beta}|\varphi_0\rangle$, given an angle $\alpha$ and any $m$-functional QADS for $U$, the probability of a positive outcome from the decision algorithm when $\beta \neq \alpha$ is*

$$DA(m, g, \beta-\alpha) := \prod_{n=0}^{m-1} \cos^2\left(g(n)\frac{\beta-\alpha}{2}\right).$$

***Proof*** Let $V$ be a unitary matrix such that $V|\varphi_0\rangle = e^{-i\alpha}U|\varphi_0\rangle = e^{i(\beta-\alpha)}|\varphi_0\rangle$. If we denote $\theta = \beta-\alpha$, we have the following probability of a positive outcome.

$$\left|\frac{1}{2^m}\sum_{x=0}^{2^m-1}\langle\varphi_0|V^B|\varphi_0\rangle\right|^2 = \left|\frac{1}{2^m}\sum_{x=0}^{2^m-1} e^{i\theta\sum x_j g(j)}\right|^2$$
$$= \left|\frac{1}{2^m}\sum_{x=0}^{2^{m-1}-1}\left(e^{i\theta\sum x_j g(j)} + e^{i\theta\sum(2^m-1-x)_j g(j)}\right)\right|^2.$$

We will now use the following property.

$$\begin{aligned} e^{ia}+e^{ib} = \cos a + \cos b + i(\sin a + \sin b) &= 2\cos\left(\frac{a+b}{2}\right)\cos\left(\frac{a-b}{2}\right) \\ &\quad +2i\sin\left(\frac{a+b}{2}\right)\cos\left(\frac{a-b}{2}\right) \\ &= 2\cos\left(\frac{a-b}{2}\right)e^{i\frac{a+b}{2}}. \end{aligned}$$

In our particular case, $a+b = \left[\sum x_j g(j) + \sum(2^m-1-x)_j g(j)\right]\theta$. Since, $x$ and $2^m-1-x$ have complementary binary digits, we get

$\left[\sum x_j g(j) + \sum (2^m - 1 - x)_j g(j)\right]\theta = \sum g(j)\theta$, and so

$$\left|\frac{1}{2^m}\sum_{x=0}^{2^{m-1}-1}\left(e^{i\theta\sum x_j g(j)} + e^{i\theta\sum(2^m-1-x)_j g(j)}\right)\right|^2 =$$

$$= \left|\frac{1}{2^m}e^{i\theta\sum g(j)}\sum_{x=0}^{2^{m-1}-1} 2\cos\left(\frac{\sum x_j g(j) - \sum(2^m-1-x)_j g(j)}{2}\theta\right)\right|^2 =$$

$$= \left|\frac{1}{2^{m-1}}\sum_{x=0}^{2^{m-1}-1}\cos\left(\frac{\sum x_j g(j) - \sum(2^m-1-x)_j g(j)}{2}\theta\right)\right|^2. \tag{A.1}$$

Next, we simplify the sum with the same procedure. We will use that $2^m - 1 - (2^{m-1} - 1 - x) = 2^m - 2^{m-1} + x = 2 \cdot 2^{m-1} - 2^{m-1} + x = 2^{m-1} + x$. Hence,

$$\sum_{x=0}^{2^{m-2}-1}\left[\cos\left(\frac{\sum x_j g(j) - \sum(2^m-1-x)_j g(j)}{2}\theta\right)\right.$$

$$\left. + \cos\left(\frac{\sum(2^{m-1}-1-x)_j g(j) - \sum(2^{m-1}+x)_j g(j)}{2}\theta\right)\right] = \sum_{x=0}^{2^{m-2}-1} 2\cos$$

$$\left(\frac{\sum x_j g(j) + \sum(2^{m-1}-1-x)_j g(j) - \left[\sum(2^m-1-x)_j g(j) + \sum(2^{m-1}+x)_j g(j)\right]}{4}\theta\right) \cdot$$

$$\cdot \cos\left(\frac{\sum x_j g(j) + \sum(2^{m-1}+x)_j g(j) - \left[\sum(2^m-1-x)_j g(j) + \sum(2^{m-1}-1-x)_j g(j)\right]}{4}\theta\right)$$

We focus on the first cosine. For $\sum x_j g(j) + \sum(2^{m-1} - 1 - x)_j g(j)$, their binary components are complementary except for the $(m-1)$-th, which is 0 for both (they are lower than $2^{m-1}$). This means that $\sum x_j g(j) + \sum(2^{m-1} - 1 - x)_j g(j) = \sum g(j) - g(m-1)$. For $\sum(2^m - 1 - x)_j g(j) + \sum(2^{m-1} + x)_j g(j)$, their binary components are complementary except for the $(m-1)$-th, which is 1 for both (they are greater than $2^{m-1}$). Therefore, $\sum(2^m - 1 - x)_j g(j) + \sum(2^{m-1} + x)_j g(j) = \sum g(j) + g(m-1)$. Hence, the first cosine is

$$\cos\left(\frac{\sum g(j) - g(m-1) - \left[\sum g(j) + g(m-1)\right]}{4}\theta\right) = \cos\left(g(m-1)\frac{\theta}{2}\right).$$

Now we focus on the second cosine. For $\sum x_j g(j) + \sum(2^{m-1} + x)_j g(j)$, they share every binary component except for the $(m-1)$-th. This implies that $\sum x_j g(j) + \sum(2^{m-1} + x)_j g(j) = 2\sum x_j g(j) - g(m-1)$. For $\sum(2^m - 1 - x)_j g(j) + \sum(2^{m-1} - 1 - x)_j g(j)$, they share every binary component except for the $(m-1)$-th. Hence, $\sum(2^m - 1 - x)_j g(j) + \sum(2^{m-1} - 1 - x)_j g(j) = 2\sum(2^{m-1} - 1 - x)_j g(j) - g(m-1)$.

Therefore, the second cosine is

$$\cos\left(\frac{2\sum x_j g(j) - g(m-1) - \left[2\sum(2^{m-1}-1-x)_j g(j) - g(m-1)\right]}{4}\theta\right) =$$
$$= \cos\left(\frac{\sum x_j g(j) - \sum(2^{m-1}-1-x)_j g(j)}{2}\theta\right).$$

Consequently, the whole sum is

$$2\cos\left(g(m-1)\frac{\theta}{2}\right)\sum_{x=0}^{2^{m-2}-1}\cos\left(\frac{\sum x_j g(j) - \sum(2^{m-1}-1-x)_j g(j)}{2}\theta\right).$$

The same process can be repeated with the remaining terms. After other $m-2$ times, we are left with

$$2^{m-1}\prod_{n=1}^{m-1}\cos\left(g(n)\frac{\theta}{2}\right)\sum_{x=0}^{0}\cos\left(\frac{\sum x_j g(j) - \sum(2-1-x)_j g(j)}{2}\theta\right) =$$
$$= 2^{m-1}\prod_{n=1}^{m-1}\cos\left(g(n)\frac{\theta}{2}\right)\cos\left(\frac{-g(0)}{2}\theta\right) = 2^{m-1}\prod_{n=0}^{m-1}\cos\left(g(n)\frac{\theta}{2}\right).$$

We only have to update the result in (A.1) to get the final formula

$$\left|\frac{1}{2^{m-1}}2^{m-1}\prod_{n=0}^{m-1}\cos\left(g(n)\frac{\theta}{2}\right)\right|^2 = \prod_{n=0}^{m-1}\cos^2\left(g(n)\frac{\beta-\alpha}{2}\right).$$

□

**Theorem 4** *Under the promise that $U|\varphi_0\rangle = e^{i\beta}|\varphi_0\rangle$ and given $\alpha$ and an $m$-geometric QADS for $U$, the probability of a positive outcome from the decision algorithm when $\beta \neq \alpha$ is*

$$DA(m, g_g, \beta-\alpha) = \frac{1-\cos(2^m(\beta-\alpha))}{2^{2m}(1-\cos(\beta-\alpha))}.$$

***Proof*** The initial probability of a positive outcome (using Corollary 1) for $\theta = \beta - \alpha$ is the geometric series

$$\left|\frac{1}{2^m}\sum_{x=0}^{2^m-1} e^{i\theta x}\right|^2 = \left|\frac{1-e^{i\theta 2^m}}{2^m(1-e^{i\theta})}\right|^2.$$

Now, we apply the following property.

$$\begin{aligned}\left|1-e^{i\gamma}\right|^2 &= |1-\cos\gamma - i\sin\gamma|^2 = (1-\cos\gamma)^2+\sin^2\gamma = 2-2\cos\gamma\\ &= 2(1-\cos\gamma).\end{aligned}$$

The result is

$$\frac{\left|1-e^{i\theta 2^m}\right|^2}{2^{2m}\left|1-e^{i\theta}\right|^2} = \frac{1-\cos(2^m(\beta-\alpha))}{2^{2m}(1-\cos(\beta-\alpha)).}$$

□

In order to prove the DA-optimality of the $m$-geometric QADS for a fixed $m$, we introduce two auxiliary results.

**Lemma 1**

$$\prod_{n=0}^{a}\cos t_n = \frac{1}{2^a}\sum_{x=0}^{2^a-1}\cos\sum_{n=0}^{a}(-1)^{x_n}t_n,$$

*where* $x = x_0 + 2x_1 + ... + 2^a x_a$ *with* $x_i \in \{0,1\}, \forall i = 0, ..., a$.

***Proof*** We will prove this result by induction on $a$. For $a = 0$, the formula is trivial: $\cos t_0 = \cos(-1)^0 t_0$. Now, we assume

$$\prod_{n=0}^{a-1}\cos t_n = \frac{1}{2^{a-1}}\sum_{x=0}^{2^{a-1}-1}\cos\sum_{n=0}^{a-1}(-1)^{x_n}t_n$$

Then,

$$\begin{aligned}\prod_{n=0}^{a}\cos t_n &= \cos t_a\prod_{n=0}^{a-1}\cos t_n = \frac{1}{2^{a-1}}\sum_{x=0}^{2^{a-1}-1}\cos t_a\cos\sum_{n=0}^{a-1}(-1)^{x_n}t_n =\\ &= \frac{1}{2^a}\sum_{x=0}^{2^{a-1}-1}\cos\left(t_a+\sum_{n=0}^{a-1}(-1)^{x_n}t_n\right)+\cos\left(t_a-\sum_{n=0}^{a-1}(-1)^{x_n}t_n\right).\end{aligned}$$

Now, we notice that $x < 2^{a-1}$ so $x_{a-1} = 0$, and if $\overline{x_i}$ is the complementary of $x_i$, then

$$\sum_{n=0}^{a-1}-(-1)^{x_n} = \sum_{n=0}^{a-1}(-1)^{\overline{x_n}} = \sum_{n=0}^{a-1}(-1)^{(2^a-1-x)_n}.$$

Applying this to the previous formula, we obtain

$$\frac{1}{2^a}\sum_{x=0}^{2^{a-1}-1}\cos\sum_{n=0}^{a}(-1)^{x_n}t_n+\cos\sum_{n=0}^{a}(-1)^{(2^a-1-x)_n}t_n.$$

And since $x$ represents all the numbers from 0 to $2^{a-1}-1$, and $2^a-1-x$ represents all the numbers from $2^{a-1}$ to $2^a-1$, it leads us to:

$$\prod_{n=0}^{a}\cos t_n=\frac{1}{2^a}\sum_{x=0}^{2^a-1}\cos\sum_{n=0}^{a}(-1)^{x_n}t_n.$$

□

Notice that $x<2^a$ implies $x_a=0$. Now, we can prove the second auxiliary result.

**Proposition 3** *Given a natural function $g:\mathbb{N}\longrightarrow\mathbb{N}$,*

$$\int_0^\pi DA(m,g,t)\,dt=\frac{\pi}{2^m}+\sum_{i=1}^{m}\frac{c_i\pi}{2^{m-1+i}},$$

*where $c_i$ is the number of equations of the form $g(n_1)\pm g(n_2)\pm\ldots\pm g(n_i)=0$, with $n_1<n_2<\ldots<n_i<m$, that $g$ satisfies.*

***Proof***

$$\int_0^\pi \mathrm{DA}(m,g,t)\,dt=\int_0^\pi\prod_{n=0}^{m-1}\cos^2\left(g(n)\frac{t}{2}\right)\,dt=\frac{1}{2^m}\int_0^\pi\prod_{n=0}^{m-1}(1+\cos g(n)t)\,dt=$$
$$=\frac{\pi}{2^m}+\frac{1}{2^m}\sum_{i=1}^{m}\sum_{j=1}^{M_i}\int_0^\pi\prod_{k=1}^{i}\cos g(n_{i,j,k})t\,dt,$$

where $M_i=\binom{m}{i}$, and each $n_{i,j,k}$ represents the $k$-th element of the $j$-th subset of $\{0,1,\ldots,m-1\}$ of size $i$ (there are $M_i$ different subsets of size $i$). Now, we can apply Lemma 1:

$$\frac{\pi}{2^m}+\sum_{i=1}^{m}\frac{1}{2^{m-1+i}}\sum_{j=1}^{M_i}\sum_{x=0}^{2^{i-1}-1}\int_0^\pi\cos\sum_{k=1}^{i}(-1)^{x_{k-1}}g(n_{i,j,k})t\,dt.$$

Since $g(n)\in\mathbb{N}$, the integrals of the cosines from 0 to $\pi$ will be 0 always, except when $\sum_{k=1}^{i}(-1)^{x_{k-1}}g(n_{i,j,k})=0$. Then, the cosines would be equal to 1, and the value of the integral equal to $\pi$. Without loss of generality, we can assume that the coefficient of the first $g(n_1)$ in each equation is 1, and therefore, for each $i$,

$$\left\{ \sum_{k=1}^{i} (-1)^{x_{k-1}} g(n_{i,j,k}) = 0 \;\middle|\; j \in [1, M_i], x \in [0, 2^{i-1} - 1] \right\} =$$
$$= \{g(n_1) \pm g(n_2) \pm \ldots \pm g(n_i) = 0, \text{ such that } n_1 < n_2 < \ldots < n_i < m\},$$

which can be straightforwardly proved by double inclusion. Finally,

$$\frac{\pi}{2^m} + \sum_{i=1}^{m} \frac{1}{2^{m-1+i}} \sum_{j=1}^{M_i} \sum_{x=0}^{2^{i-1}-1} \int_0^{\pi} \cos \sum_{k=1}^{i} (-1)^{x_{k-1}} g(n_{i,j,k}) t \, dt = \frac{\pi}{2^m} + \sum_{i=1}^{m} \frac{c_i \pi}{2^{m-1+i}},$$

where $c_i$ is the number of equations of the form $g(n_1) \pm g(n_2) \pm \ldots \pm g(n_i) = 0$, with $n_1 < n_2 < \ldots < n_i < m$, that $g$ satisfies. □

We are now ready to prove the final theorem.

**Theorem 5** *The $m$-geometric QADS is DA-optimal for any fixed $m$. Moreover, among those $m$-functional QADS which are DA-optimal for $m$, $m$-geometric QADS have the smallest size.*

***Proof*** Using Proposition 3, the optimality is almost immediate just by noticing that, in the case of an $m$-geometric QADS, $c_i = 0, \forall i$.

If we try to build a DA-optimal algorithm with the smallest possible size, we start by $g(0) = 1$. We cannot set $g(0) = 0$, since that would imply $c_1 > 0$. Now, $g(1)$ must be different in order to avoid that $g(0) - g(1) = 0$, so the smallest choice is $g(1) = 2$. In order to avoid that $g(0) + g(1) - g(2) = 0$, we must take $g(2) = 4$. If we keep the process going, we obtain the $m$-geometric QADS. □

### A.3.: Results in Sect. 5.

In order to prove Theorem 6, we will start by a useful lemma.

**Lemma 2** *Under the conditions of Theorem 6, for $\theta_k = \lambda_k - \alpha$ and $\psi_k = \psi_k = |\langle \varphi_k | \psi \rangle|^2$,*

$$\widetilde{DA}(m, g, |\psi\rangle, U, \alpha) = \left| \sum_{k=0}^{2^d-1} e^{i\theta_k \sum g(j)} \sqrt{DA(m, g, \theta_k)} \psi_k \right|^2 .$$

***Proof*** We will apply $V = e^{-i\alpha} U = \sum_{k=0}^{2^d-1} e^{-i\theta_k} |\varphi_k\rangle\langle\varphi_k|$ in the circuit so, again, the probability of a positive outcome is

$$\left|\frac{1}{2^m}\sum_{x=0}^{2^m-1}\langle\psi|V^B|\psi\rangle\right|^2=\left|\frac{1}{2^m}\sum_{x=0}^{2^m-1}\sum_{k=0}^{2^d-1}\langle\psi|\left(e^{i\theta_k B}|\varphi_k\rangle\langle\varphi_k|\right)|\psi\rangle\right|^2$$
$$=\left|\frac{1}{2^m}\sum_{k=0}^{2^d-1}\left(\sum_{x=0}^{2^m-1}e^{i\theta_k\sum x_j g(j)}\right)\psi_k\right|^2.$$

The rest of the proof is easily deduced by repeating the exact same process of Theorem 3 but without ever applying $|\cdot|^2$. □

**Theorem 6** *In the DA for geometric QADS, suppose $|\varphi_0\rangle$ is the exact eigenvector of $U$, with eigenvalue $e^{i\beta}$, that we are interested in, but the initial inexact state $|\psi\rangle$ satisfies $|\langle\varphi_0|\psi\rangle|^2=1-\delta$. If $\widetilde{DA}(m,g_g,|\psi\rangle,U,\alpha)$ is the new probability of error of the DA initialized in $|\psi\rangle$, then:*

- *If $DA(m,g_g,\beta-\alpha)\leq\widetilde{DA}(m,g_g,|\psi\rangle,U,\alpha)$, then*
$$\left|DA(m,g_g,\beta-\alpha)-\widetilde{DA}(m,g_g,|\psi\rangle,U,\alpha)\right|<\delta^2. \tag{A.2}$$
- *If $DA(m,g_g,\beta-\alpha)>\widetilde{DA}(m,g_g,|\psi\rangle,U,\alpha)$, then*
$$\left|DA(m,g_g,\beta-\alpha)-\widetilde{DA}(m,g_g,|\psi\rangle,U,\alpha)\right|<4(\delta-\delta^2). \tag{A.3}$$

***Proof*** We are assuming $\langle\varphi_0|\psi\rangle^2=\psi_0=1-\delta$, and we know that $\sum\psi_k=1$, so this is a summary of the available information.

$$\psi_0=1-\delta, \tag{A.4}$$
$$\sum_{k=1}^{2^d-1}\psi_k=\delta, \tag{A.5}$$
$$\sum_{k=1}^{2^d-1}\psi_k^2\leq\delta^2, \tag{A.6}$$
$$0\leq \mathrm{DA}(m,g,t)\leq 1. \tag{A.7}$$

The next step is to apply Lemma 2 for geometric QADS. For case 1, we proceed as follows.

$$\widetilde{\mathrm{DA}}(m,g_g,|\psi\rangle,U,\alpha)=\left|\sum_{k=0}^{2^d-1}e^{i\theta_k(2^m-1)}\sqrt{\mathrm{DA}(m,g_g,\theta_k)}\psi_k\right|^2\leq$$
$$\leq\sum_{k=0}^{2^d-1}\left|e^{i\theta_k(2^m-1)}\sqrt{\mathrm{DA}(m,g_g,\theta_k)}\right|^2\psi_k^2=\sum_{k=0}^{2^d-1}\mathrm{DA}(m,g_g,\theta_k)\psi_k^2 \tag{A.8}$$

Now, if $\mathrm{DA}(m, g_g, \beta-\alpha) \leq \widetilde{\mathrm{DA}}(m, g_g, |\psi\rangle, U, \alpha)$, then

$$\begin{aligned}
&\left|\mathrm{DA}(m, g_g, \beta-\alpha) - \widetilde{\mathrm{DA}}(m, g_g, |\psi\rangle, U, \alpha)\right| \\
&= \widetilde{\mathrm{DA}}(m, g_g, |\psi\rangle, U, \alpha) - \mathrm{DA}(m, g_g, \beta-\alpha) \leq \\
&\leq \sum_{k=0}^{2^d-1} \mathrm{DA}(m, g_g, \theta_k)\psi_k^2 - \mathrm{DA}(m, g_g, \beta-\alpha) \\
&= \sum_{k=1}^{2^d-1} \mathrm{DA}(m, g_g, \theta_k)\psi_k^2 - \mathrm{DA}(m, g_g, \beta-\alpha)(1-\psi_0^2).
\end{aligned}$$

Applying A.6 and A.7, we obtain

$$\sum_{k=1}^{2^d-1} \mathrm{DA}(m, g_g, \theta_k)\psi_k^2 - \mathrm{DA}(m, g_g, \beta-\alpha)(1-\psi_0^2) \leq \sum_{k=1}^{2^d-1} \psi_k^2 - 0(1-\psi_0^2) \leq \delta^2. \tag{A.9}$$

However, since in (A.8) the equality is only reached when $\lambda_i = \lambda_j, \forall i, j$, but in (A.9) the equality is only reached when either $\lambda_k = \beta + 2q\pi/2^m \neq \beta$, for all $k \neq 0$; or $\lambda_k = \beta - 2q\pi/2^m \neq \beta$, for all $k \neq 0$ and for some $q \in \{1, \ldots, 2^m - 1\}$, both conditions cannot be met simultaneously. Therefore,

$$\left|\mathrm{DA}(m, g_g, \beta-\alpha) - \widetilde{\mathrm{DA}}(m, g_g, |\psi\rangle, U, \alpha)\right| < \delta^2.$$

For the second bound, $\widetilde{\mathrm{DA}}(m, g_g, |\psi\rangle, U, \alpha)$ is going to appear with a negative sign, so we have to obtain a lower bound. We start by rewriting it.

$$\begin{aligned}
&\widetilde{\mathrm{DA}}(m, g_g, |\psi\rangle, U, \alpha) = \left|\sum_{k=0}^{2^d-1} \left(e^{i\theta_k(2^m-1)} \prod_{n=0}^{m-1} \cos\theta_k 2^{n-1}\right)\psi_k\right|^2 = \\
&= \left|e^{i(\beta-\alpha)(2^m-1)}\sqrt{\mathrm{DA}(m, g_g, \beta-\alpha)}\psi_0 + \sum_{k=1}^{2^d-1} e^{i\theta_k(2^m-1)}\sqrt{\mathrm{DA}(m, g_g, \theta_k)}\psi_k\right|^2.
\end{aligned}$$

Our objective is to make this value as low as possible in order to find the error of the worst case. We already have a vector, $e^{i(\beta-\alpha)(2^m-1)}$, that we cannot adjust, ‘pushing’ towards one direction. So, under the natural assumption of $\delta < 1/2$, our best shot for decreasing the vector length is to make every other vector ‘pull’ in the opposite direction. That is, for all $k$,

$$(\lambda_k - \alpha)(2^m - 1) = (\beta - \alpha)(2^m - 1) \pm \pi \Leftrightarrow \lambda_k = \beta \pm \frac{\pi}{2^m - 1}. \tag{A.10}$$

Also, we substitute the associated coefficients by 1, their highest possible value. Then, by virtue of (A.4) and (A.5), we obtain

$$\left| e^{i(\beta-\alpha)(2^m-1)}\sqrt{\mathrm{DA}(m, g_g, \beta-\alpha)}\psi_0 + \sum_{k=1}^{2^d-1} e^{i\theta_k(2^m-1)}\sqrt{\mathrm{DA}(m, g_g, \theta_k)}\psi_k \right|^2 \geq$$

$$\geq \left| e^{i(\beta-\alpha)(2^m-1)}\sqrt{\mathrm{DA}(m, g_g, \beta-\alpha)}\psi_0 - e^{i(\beta-\alpha)(2^m-1)}\sum_{k=1}^{2^d-1}\psi_k \right|^2 =$$

$$= \left(\sqrt{\mathrm{DA}(m, g_g, \beta-\alpha)}(1-\delta) - \delta\right)^2.$$

Notice that both terms of the subtraction have turned into positive real numbers. Now, if $\mathrm{DA}(m, g_g, \beta-\alpha) > \widetilde{\mathrm{DA}}(m, g_g, |\psi\rangle, U, \alpha)$. Then

$$\begin{aligned}
&\left|\mathrm{DA}(m, g_g, \beta-\alpha) - \widetilde{\mathrm{DA}}(m, g_g, |\psi\rangle, U, \alpha)\right| \\
&= \mathrm{DA}(m, g_g, \beta-\alpha) - \widetilde{\mathrm{DA}}(m, g_g, |\psi\rangle, U, \alpha) \leq \\
&\leq \mathrm{DA}(m, g_g, \beta-\alpha) - \left(\sqrt{\mathrm{DA}(m, g_g, \beta-\alpha)}(1-\delta) - \delta\right)^2 = \\
&= \mathrm{DA}(m, g_g, \beta-\alpha) - \mathrm{DA}(m, g_g, \beta-\alpha)(1-2\delta+\delta^2) \\
&\quad +2(\delta-\delta^2)\sqrt{\mathrm{DA}(m, g_g, \beta-\alpha)} - \delta^2 = \\
&= 2\left(\mathrm{DA}(m, g_g, \beta-\alpha) + \sqrt{\mathrm{DA}(m, g_g, \beta-\alpha)}\right)\delta - \left(\sqrt{\mathrm{DA}(m, g_g, \beta-\alpha)}+1\right)^2\delta^2.
\end{aligned}$$

If we define $f(x) = 2(x+\sqrt{x})\delta - (\sqrt{x}+1)^2\delta^2$, then $f'(x) = 0$ if and only if

$$2\delta\sqrt{x} + \delta - \delta^2\sqrt{x} - \delta^2 = 0 \Leftrightarrow (2-\delta)\sqrt{x} = \delta - 1 \Leftrightarrow \sqrt{x} = \frac{\delta-1}{2-\delta} < 0 \ \#$$

This means that the maximum of this expression is found in one of the endpoints. In this case, for $\mathrm{DA}(m, g_g, \beta-\alpha) = 1$. Therefore,

$$\begin{aligned}
&2\left(\mathrm{DA}(m, g_g, \beta-\alpha) + \sqrt{\mathrm{DA}(m, g_g, \beta-\alpha)}\right)\delta \\
&-\left(\sqrt{\mathrm{DA}(m, g_g, \beta-\alpha)}+1\right)^2\delta^2 \leq 4(\delta-\delta^2).
\end{aligned}$$

However, we deduced this bound by assuming that $\mathrm{DA}(m, g_g, \beta-\alpha) = \mathrm{DA}(m, g_g, \lambda_k-\alpha) = 1$, which is only possible if $\beta = \alpha = \lambda_k$. But, if that was the case, then from the equation in Lemma 2 it is easy to see that $\widetilde{\mathrm{DA}}(m, g_g, |\psi\rangle, U, \alpha) = \mathrm{DA}(m, g_g, \beta-\alpha)$ and the error would be 0. Thus, the bound is never reached and

$$\left|\mathrm{DA}(m, g_g, \beta-\alpha) - \widetilde{\mathrm{DA}}(m, g_g, |\psi\rangle, U, \alpha)\right| < 4(\delta-\delta^2).$$

□

In order to study the second bound further, let us come back to (A.10). But now, the option that makes $\mathrm{DA}\Big(m, g_g, \beta-\alpha \pm \frac{\pi}{2^m-1}\Big)$ greater for each $\alpha$ will be called $\lambda(\alpha)$. Then we obtain

$$\left|e^{i(\beta-\alpha)(2^m-1)}\sqrt{\mathrm{DA}(m, g_g, \beta-\alpha)}\psi_0 + \sum_{k=1}^{2^d-1} e^{i\theta_k(2^m-1)}\sqrt{\mathrm{DA}(m, g_g, \theta_k)}\psi_k\right|^2 \geq$$
$$\geq \left(\sqrt{\mathrm{DA}(m, g_g, \beta-\alpha)(1-\delta)} - \sqrt{\mathrm{DA}(m, g_g, \lambda(\alpha)-\alpha)\delta}\right)^2.$$

And, following the same procedure,

$$\begin{aligned}&2\left|\mathrm{DA}(m, g_g, \beta-\alpha) - \widetilde{\mathrm{DA}}(m, g_g, |\psi\rangle, U, \alpha)\right| \leq \\ &2 \leq 2\left(\mathrm{DA}(m, g_g, \beta-\alpha) + \sqrt{\mathrm{DA}(m, g_g, \beta-\alpha)\mathrm{DA}(m, g_g, \lambda(\alpha)-\alpha)}\right)\delta - \\ &2 - \left(\sqrt{\mathrm{DA}(m, g_g, \beta-\alpha)} + \sqrt{\mathrm{DA}(m, g_g, \lambda(\alpha)-\alpha)}\right)^2 \delta^2.\end{aligned}$$

A numerical study shows that the maximum of this expression is reached really close to $\beta-\alpha = \pi/2^{m+2}$. Then, from Theorem 4 we can deduce that

$$\lim_{m\to\infty} \mathrm{DA}\left(m, g_g, \frac{\pi}{2^{m+2}}\right) = \frac{16(2-\sqrt{2})}{\pi^2},$$
$$\lim_{m\to\infty} \mathrm{DA}\left(m, g_g, \frac{\pi}{2^{m+2}-\frac{\pi}{2^m-1}}\right) = \frac{16(2+\sqrt{2})}{9\pi^2}.$$

Also, the limit is approached from below. This means that the bound for this point would be approximately $3.428\delta - 3.093\delta^2$.

## Appendix B: Design rationale of the practical applications

### B.1.: Decision on an interval

In order to calculate the probability of error of this algorithm, notice that the DA follows a Bernoulli distribution. Therefore, the distribution of its expected value $P_\alpha$ can be approximated by a normal distribution:

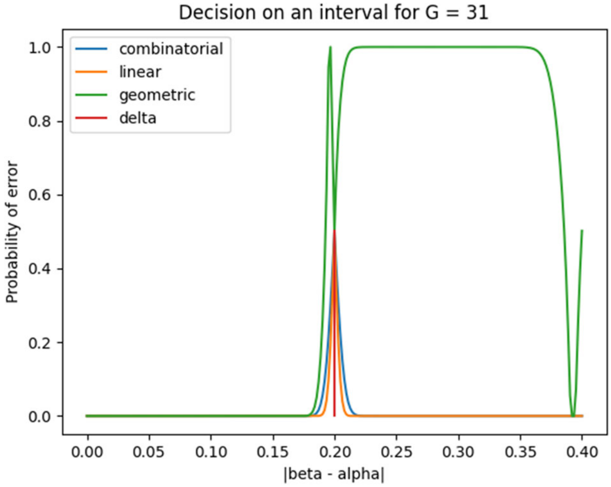


**Fig. 22** Probability of error of the decision on an interval algorithm for the combinatorial, linear and geometric QADS for a fixed $G = 31$, sample size of 1000 and $\delta = 0.2$

$$\frac{\overline{X}_n - \mu}{\sigma/\sqrt{n}} = \frac{\overline{X}_n - p}{\sqrt{p(1-p)/n}} \xrightarrow[n\to\infty]{} \mathcal{N}(0, 1).$$

Using this approximation, we compute the probability of error of the whole algorithm for any $\beta - \alpha$, as shown in Figs. 8 and 9.

Sometimes, when $m$ is not suitably chosen depending on $\delta$, the geometric QADS may develop unacceptable probabilities of error. This is due to the fact that $\mathrm{DA}(m, g_g, t)$, as shown in Fig. 7, is not decreasing on every point, and $P_\delta$ might be greater than $P_\alpha$, despite $\beta$ being in $[\alpha - \delta, \alpha + \delta]$. An example of this behaviour can be seen in Fig. 22. A simple solution is the use of linear QADS whenever the necessary conditions for the geometric QADS to work correctly cannot be verified with certainty. In this direction, notice that the geometric QADS gives 0 in every point of the form $2\pi/2^m$, and $P_\delta$ should be greater than the probability on the first peak, which is near

$$|\beta - \alpha| = \frac{2\pi}{2^m} + \frac{2\pi}{2^{m+1}} = \frac{3\pi}{2^m}$$

So, finding the greatest $m$ such that $P_\delta$ is well greater than this probability, would provide the smallest error probability for the geometric QADS.

### B.2.: Interval correction (IC)

In this algorithm, the case $P_\alpha \approx P_\delta$ is treated differently, assuming that $\beta$ is near one of the endpoints of the interval. Such an endpoint is found by selecting a geometric

QADS, and a proper $m_g$ such that

$$\frac{2\pi}{2^{m_g}} \approx 2\delta \Leftrightarrow m_g \approx \log_2\left(\frac{\pi}{\delta}\right).$$

With this choice the probability of a positive outcome on the correct endpoint is near 1, and the probability of a negative outcome on the incorrect endpoint is near 0, so the right endpoint will be probably found with just one DA. Anyway, the DA can be repeated on both endpoints for extra confidence. This procedure considerably improves the probability of error, while adding two extra runs of a DA. With this new approach, we will consider two reference probabilities. Let us take $d_1 < \delta$ and $d_2 > \delta$. We will apply the previous strategy whenever $P_\alpha$ falls between both reference probabilities: $P_{d_1} := \mathrm{DA}(m, g, d_1)$ and $P_{d_2} := \mathrm{DA}(m, g, d_2)$. The selection of $d_1$ and $d_2$ (which may yield remarkable differences) will be discussed later, as the closer they are, the bigger the probability of error around $|\beta - \alpha| = \delta$. We can proceed now to calculate the probability of error for the three possible cases.
*Case 1*: $|\beta - \alpha| \leq \delta$. The probability of error will have two independent components. First, when $P_\alpha \leq P_{d_2}$, because then the algorithm would discard the interval. As we already noticed,

$$P_\alpha \xrightarrow[n\to\infty]{} \mathcal{N}\left(p, \sqrt{\frac{p(1-p)}{n}}\right),$$

being $p = \mathrm{DA}(m, g, \beta - \alpha)$. Second, when $P_{d_2} \leq P_\alpha \leq P_{d_1}$, and the change of interval leaves $\beta$ outside. This happens when the geometric QADS-based DA gives the wrong answer on both endpoints of the interval. Because of Theorem 4, the error probabilities at the endpoints would be

$$1 - \frac{1 - \cos(2^m(\delta - |\beta - \alpha|))}{2^{2m}(1 - \cos(\delta - |\beta - \alpha|)} \text{ and } \frac{1 - \cos(2^m(\delta + |\beta - \alpha|))}{2^{2m}(1 - \cos(\delta + |\beta - \alpha|))}.$$

The total probability of error in this case is

$$P(P_\alpha \leq P_{d_2}) + P(P_{d_2} < P_\alpha \leq P_{d_1}) \cdot \left(1 - \frac{1 - \cos(2^m(\delta - |\beta - \alpha|))}{2^{2m}(1 - \cos(\delta - |\beta - \alpha|))}\right)$$
$$\cdot \frac{1 - \cos(2^m(\delta + |\beta - \alpha|))}{2^{2m}(1 - \cos(\delta + |\beta - \alpha|))}$$

*Case 2*: $\delta < |\beta - \alpha| \leq 2\delta$. The probability of error in this case is analogous to the Case 1, except for the first error probability. Here, $P_\alpha > P_{d_1}$ would yield a wrong acceptance of the given interval.

$$P(P_\alpha > P_{d_1}) + P(P_{d_2} < P_\alpha \leq P_{d_1}) \cdot \left(1 - \frac{1 - \cos(2^m(\delta - |\beta - \alpha|))}{2^{2m}(1 - \cos(\delta - |\beta - \alpha|))}\right)$$

$$\cdot \frac{1-\cos(2^m(\delta+|\beta-\alpha|))}{2^{2m}(1-\cos(\delta+|\beta-\alpha|))}$$

*Case 3*: $2\delta < |\beta - \alpha|$. Here, if $P_{d_2} \le P_\alpha \le P_{d_1}$, no matter which is the new interval, it will not contain $\beta$ anymore. Therefore, the probability of error is just $P(P_\alpha > P_{d_2})$.

Finally, let us discuss the best choice of $d_1$ and $d_2$ in order to minimize the maximum of the error probability. Let us write $d_1 = c\delta$, where $c \in (0, 1)$, and take $P_{d_2} = 2P_\delta - P_{d_1}$. This choice ensures the existence of the limit of the previously discussed probability of error when $t \to \delta$, so that the function is continuous on $t = \delta$. In other words, it ensures that $P(P_\alpha > P_{d_1}) = P(P_\alpha \le P_{d_2})$ when $|\beta - \alpha| = \delta$, correctly linking cases 1 and 2.

Now, $c$ will be adjusted up to two decimals. Notice that, for $c = 1$, the maximum probability of error occurs at $t = \delta$, and it decreases with smaller values of $c$. So, we just have to find the greatest $c$ such that this maximum is not at $t = \delta$. All of this setting does not depend on $\beta$ and hence can be prepared classically. As above, the geometric QADS will yield wrong outputs when $\delta$ and $m$ are not correctly chosen.

### B.3.: $\delta$-Approximation algorithm

The algorithm consists of two parts: finding the first $[\alpha_0 - \delta_0, \alpha_0 + \delta_0]$, and checking intervals of decreasing length until one of the form $[\alpha - \delta, \alpha + \delta]$ is found. Let us give the details of both.

*Selecting the initial interval.* It is desirable to get $\alpha_0$ close to $\beta$ at the beginning. To do this, any phase estimation method can be used, but we will stick to the Hadamard test due to its direct relation with functional QADS. We run it with $m = 1$, and $h_0$ repetitions of the DA. However, if $\beta > \pi$, the Hadamard test will actually approximate $-\beta$. So, once a first approximation $\alpha_0$ is obtained, a decision between $\alpha_0$ and $-\alpha_0$ should be taken, by an application of a geometric QADS, just like was made in the IC for a choice between $\alpha - \delta$ and $\alpha + \delta$. In this case:

$$m_g \approx \log_2\left(\frac{\pi}{\alpha_0}\right), \quad \text{if } \alpha_0 \le \frac{\pi}{2};$$
$$m_g \approx \log_2\left(\frac{\pi}{\pi-\alpha_0}\right), \quad \text{if } \alpha_0 > \frac{\pi}{2}.$$

We can also set a maximum for $m_g$ since, in case $\alpha_0$ is really close to 0 or $\pi$ (and so to $-\alpha_0$, too), it is not worth investing a lot of qubits and operations to choose between them.

For $\delta_0$, we must decide how many intervals we want to split the domain into, $n_0$, on each of the iterations of the algorithm, and select $\delta_0 = n_0^{t-1}\delta$, where $t$ represents the number of iterations of the algorithm. Moreover, $t$ is chosen so that the length of the initial interval is approximately, $2\pi/n_0$:

$$2\delta_0 = 2 \cdot n_0^{t-1}\delta \approx \frac{2\pi}{n_0} \;\Leftrightarrow\; t \approx \log_{n_0}\left(\frac{\pi}{\delta}\right).$$

All of the $m$'s, $\delta_0$'s and $c$'s that are going to be used in each iteration of the algorithm are computed classically at the beginning.

*Evolving the selected interval.* Now that we have the initial $[\alpha_0 - \delta_0, \alpha_0 + \delta_0]$, we repeatedly apply the IC $n$ times to it. We will use geometric QADS with $m$, $P_{d_1}$ and $P_{d_2}$ adjusted as established before, as this minimizes the probability of error for the algorithm.

Since we assume that $\alpha_0$ is a nice approximation of $\beta$, if the IC outputs a negative result, we keep trying intervals surrounding $\alpha_0$. For example, suppose $\alpha_i = 0$ and $\delta_i = 1$. If $i = 0$, the algorithm would check $[-1, 1], [0, 2], [-2, 0], [1, 3], [-3, -1], \ldots$; if $i \neq 0$, it would check $[-1, 1], [1, 3], [-3, -1], [3, 5], [-5, -3], \ldots$. In general, in the first iteration, after $z$ negative results, we check the interval $[A - \delta_0, A + \delta_0]$, where

$$\begin{aligned} A &= \alpha_0 + \frac{z+1}{2}\delta_0, \quad \text{if } z \text{ odd;} \\ A &= \alpha_0 - \frac{z}{2}\delta_0, \quad \text{if } z \text{ even.} \end{aligned}$$

For the rest of iterations,

$$\begin{aligned} A &= \alpha_i + (z+1)\delta_i, \quad \text{if } z\text{odd;} \\ A &= \alpha_i - z\delta_i, \quad \text{if } z \text{ even.} \end{aligned}$$

The more thorough search in the first iteration successfully avoids: (1) missing the correct interval, and (2) giving a positive outcome on a wrong interval. Solutions to both problems usually imply a huge number of extra operations. Another way of dealing with (2), is to set a maximum number of failed IC on each iteration so that, if reached, we backtrack to the previous iteration.

Once an IC decides $\beta \in [A - \delta_i, A + \delta_i]$ (maybe after a correction), we update $\delta_{i+1} = \delta_i / n_0$, $\alpha_{i+1} = A$, and proceed to the next iteration.

The choice of $h_0$, $n_0$ and $n$ can affect the probability of error and the number of operations, so we can combine them as desired. We established $n = 30$ and, for $\delta \approx 0.01$, taking $h_0$ around 6000 and $n_0$ around 15 seemed to be the most balanced approach. Of course, the whole algorithm might not be fully optimal and more improvements might be possible.

## References

1. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, STOC '96, ACM, NY, USA (1996), pp. 212–219

2. Combarro, E.F., Ranilla, J., Rúa, I.F.: Quantum abstract detecting systems. Quantum Inf. Process. **19**(8), 258 (2020)
3. Deutsch, D., Jozsa, R.: Rapid solution of problems by quantum computation. Proc. R. Soc. Lond. A Math. Phys. Eng. Sci. **439**(1907), 553–558 (1992)
4. Ambainis, A., Kempe, J, Rivosh, A.: Coins make quantum walks faster. In: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '05, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2005), pp. 1099–1108
5. Portugal, R.: Quantum Walks and Search Algorithms. Springer, New York (2013)
6. Santos, R.A.M.: Szegedy's quantum walk with queries. Quantum Inf. Process. **15**(11), 4461–4475 (2016)
7. Szegedy, M.: Quantum speed-up of markov chain based algorithms. In: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, FOCS '04, IEEE Computer Society, Washington, DC, USA, (2004) pp. 32–41
8. Wong, T.: Equivalence of Szegedy's and coined quantum walks. Quantum Inf Process **16**, 215 (2017)
9. Combarro, E. F., Ranilla, J., Rúa, I.: A quantum algorithm for the commutativity of finite dimensional algebras, , IEEE Access **7**, 45554–45562 (2019)
10. Lugilde Fernández, G., Combarro, E.F., Rúa, I.F.: Quantum measurement detection algorithms. Quantum Inf. Process. **21**(8), 274 (2022)
11. Hernández Cáceres, I.R.J.M., Combarro, E.F.: Combinatorial and rotational quantum abstract detecting systems. Quantum Inf. Process. **21**(2), 1–27 (2022)
12. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information, 10th Anniversary Cambridge University Press, Cambridge (2011)
13. Bolstad, W.M., Curran, J.M.: Introduction to Bayesian Statistics. John Wiley & Sons (2016)
14. O'Brien, T.E., Tarasinski, B., Terhal, B.M.: Quantum phase estimation of multiple eigenvalues for small-scale (noisy) experiments. New J. Phys. **21**(2), 023022 (2019)
15. Mohammadbagherpoor, H., Oh, Y.-H., Singh, A., Yu, X., Rindos, A. J.: Experimental challenges of implementing quantum phase estimation algorithms on IBM quantum computer. arXiv preprint arXiv:1903.07605
16. Nakaji, K.: Faster amplitude estimation. arXiv preprint arXiv:2003.02417
17. Rall, P.: Faster coherent quantum algorithms for phase, energy, and amplitude estimation. Quantum **5**, 566 (2021)
18. Suzuki, Y., Uno, S., Raymond, R., Tanaka, T., Onodera, T., Yamamoto, N.: Amplitude estimation without phase estimation. Quantum Inf. Process. **19**, 1–17 (2020)
19. Najafi, P., Costa, P., Berry, D.W.: Optimum phase estimation with two control qubits. AVS Quantum Sci. **5**, (2) (2023)
20. O'Loan, C. J.: Iterative phase estimation. arXiv preprint arXiv:0904.3426 43 (1)
21. Kitaev, A.Y., Shen, A., Vyalyi, M.N.: Classical and Quantum Computation, vol. 47. American Mathematical Society, Providence (2002)
22. Svore, K. M., Hastings, M. B., Freedman, M.: Faster phase estimation. arxiv preprint arXiv:1304.0741