Quantum Information and Computation, Vol. 14, No. 1&2 (2014) 0056–0090

QUANTUM ALGORITHMS FOR ONE-DIMENSIONAL INFRASTRUCTURES

PRADEEP SARVEPALLI
Department of Electrical Engineering, IIT Madras
Chennai, Tamilnadu 600 036, India

PAWEL WOCJAN
Department of Electrical Engineering and Computer Science, University of Central Florida
Orlando, FL 32816, USA

Received June 11, 2012
Revised May 3, 2013

Infrastructures are group-like objects that make their appearance in arithmetic geometry in the study of computational problems related to number fields and function fields over finite fields. The most prominent computational tasks of infrastructures are the computation of the circumference of the infrastructure and the generalized discrete logarithms. Both these problems are not known to have efficient classical algorithms for an arbitrary infrastructure. Our main contributions are polynomial time quantum algorithms for one-dimensional infrastructures that satisfy certain conditions. For instance, these conditions are always fulfilled for infrastructures obtained from number fields and function fields, both of unit rank one. Since quadratic number fields give rise to such infrastructures, this algorithm can be used to solve Pell's equation and the principal ideal problem. In this sense we generalize Hallgren's quantum algorithms for quadratic number fields, while also providing a polynomial speedup over them. Our more general approach shows that these quantum algorithms can also be applied to infrastructures obtained from complex cubic and totally complex quartic number fields. Our improved way of analyzing the performance makes it possible to show that these algorithms succeed with constant probability independent of the problem size. In contrast, the lower bound on the success probability due to Hallgren decreases as the fourth power of the logarithm of the circumference. Our analysis also shows that fewer qubits are required. We also contribute to the study of infrastructures, and show how to compute efficiently within infrastructures.

Keywords: quantum algorithms, infrastructures, circumference, period finding, quantum Fourier transform, discrete log problem

Communicated by: R Jozsa & M Mosca

1 Introduction

One of the most important challenges in quantum computing has been the task of finding efficient algorithms for problems that are intractable on a classical computer. Following Shor's discovery of a polynomial time quantum algorithm for factoring integers and solving the discrete logarithm problem [1], the key ideas of the period finding algorithm were generalized

and led to the framework of the hidden subgroup problem (HSP) [2]. The major algorithmic success in this context is that the abelian HSP can be solved efficiently by a quantum algorithm (while classical algorithms are inefficient). This quantum algorithm can also be viewed as determining the structure of a hidden lattice Λ inside $\mathbb{Z}^n$.

An important restriction of this quantum algorithm is that it only works for integral lattices. But, Hallgren overcame this obstacle in the one-dimensional setting by generalizing Shor's period finding algorithm to the case where the period is irrational [3,4] (see also [5,6]). This enabled him to give polynomial time quantum algorithms for computing the regulator of a quadratic number field and solving the principal ideal problem. Schmidt and Vollmer [7,8] and Hallgren [9] presented a polynomial time quantum algorithm for determining a hidden lattice in $\mathbb{R}^n$ for fixed n. They showed that computing the unit group and solving the principal ideal problem in number fields of fixed unit rank can be solved efficiently with this algorithm.[a] In contrast to $\mathbb{Z}^n$, the success probability of the above quantum algorithms for finding a hidden lattice in $\mathbb{R}^n$ decreases exponentially with the dimension, making them inefficient with respect to the dimension. Thus, an important open problem is to determine whether there exist quantum algorithms whose success probability decrease less rapidly with the dimension.

In this paper, we initiate the study of quantum algorithms for infrastructures. These group-like structures are hidden beneath the number theoretic details of the above quantum algorithms. They play an important role in the research on computational problems in global fields, i.e. number fields and function fields over finite fields [10] (arithmetic geometry provides a unified treatment of global fields [11]). For instance, computing the unit group and solving the principal ideal problem can both be translated to well defined problems of infrastructures, namely, the computation of the lattice characterizing the periodic symmetry of the infrastructure and the computation of generalized discrete logarithms in these group-like structures. Both these computational problems associated with the infrastructures are not known to have efficient classical algorithms.

We focus here on arbitrary one-dimensional infrastructures and give polynomial time quantum algorithms for computing the circumference and for computing the generalized discrete logarithms. One-dimensional infrastructures arise from global fields of unit rank, and include the special case of real quadratic number fields studied by Hallgren [4] and complex cubic and quartic number fields [12]. Our algorithms perform better than the algorithms of [4] when applied to these problems. The proposed algorithms achieve a super polynomial speedup over the best known classical algorithms.

In summary, we make the following contributions:

- Firstly, although our algorithms are given in a more general setting, they have a lower time and also space complexity than those in [3,4]. Denote by $g_H(m)$, $s_H(m)$, and $p_H(m)$ the gate complexity (number of gates), the space complexity (number of qubits), and the guaranteed success probability of the algorithms in [3,4], where $m = \log M$ and M is an upper bound on the product of the circumference of the infrastructure and the reciprocal of the minimum distance between two adjacent elements of the infrastructure.[b]

[a] Hallgren also showed in [9] how to compute the class group of a number field of fixed unit rank.
[b] The success probability can obviously be boosted arbitrarily close to 1 at the cost of increasing time complexity

Similarly, let $g(m), s(m)$, and $p(m)$ denote the gate complexity (number of gates), the space complexity (number of qubits), and the guaranteed success probability of our algorithms.

We have $g_H(m) \geq g(m)$ and $s_H(m) \geq s(m)$, while $p_H(m) \leq p(m)$. We have $p_H(m) \leq 10^{-9}$ and $p_H(m)$ decreases as m^{-4}, whereas $p(m) \geq 10^{-5}$ is bounded from below by a constant for all m. The success probability of a quantum algorithm due to Schmid [6] is bounded from below by 2^{-26}.

- Secondly, our results when specialized to quadratic number fields provide a simpler treatment of the computational problems, and can be easily applied without extensive knowledge of number theory.

- Thirdly, we introduce an interesting technical result (Lemma 18) that could have wider applicability in the analysis of quantum algorithms employing quantum Fourier transform.

- Finally, we make a contribution to the study of one-dimensional infrastructures by showing how to perform finite precision computations efficiently within the infrastructures. These are useful even in the context of purely classical algorithms for infrastructures.

A natural direction for further investigation is the generalization of the proposed quantum algorithms for higher dimensional infrastructures. These are presented in [13].

This paper is structured as follows. We first introduce the mathematical preliminaries, defining precisely the notion of an infrastructure and the computational problems associated with them. We then show that these infrastructures can be endowed with a group structure and review the relevant results related to the embedding of the infrastructures into circle groups. We then introduce group homomorphisms that are key to solving the computational problems associated to them. We also show that these homomorphisms can be computed efficiently. These results should be of interest beyond the present context.

In section 3, we generalize the notion of periodic quantum states and prove a key technical result related to the analysis of Fourier sampling. This result simplifies the analysis of the algorithms and leads to a tighter bound on the success probabilities of the proposed algorithms. In this section, we give a quantum algorithm for estimating the period of a pseudo-periodic quantum state. This result could be applicable to situations beyond the current setting of infrastructures.

In section 4, we show how to set up periodic quantum states from infrastructures and use the quantum algorithm proposed in section 3 to estimate the circumference of the infrastructure. In section 5, we present the quantum algorithm to solve the generalized discrete logarithm problem.

---

by repeating the original algorithm multiple times and checking each time if the output is correct. Therefore, we need to say in more detail what we mean by $p_H(m)$ and $p(m)$. The guaranteed success probabilities $p_H(m)$ and $p(m)$ above correspond to the case that the Fourier sampling process has been only performed twice (or equivalently, that only two so-called periodic states have been prepared.)

2 Infrastructures

We define infrastructures and state the two main computational problems associated to infrastructures. We restrict our attention to the one-dimensional infrastructures.

2.1 Definition of infrastructures

We refer the reader to [10, 14, 15] for more information on infrastructures. Our presentation follows [14, 16].

Definition 1: (Infrastructure) An infrastructure of circumference R is a pair (X, d) where X is a finite set and $d : X \hookrightarrow \mathbb{R}/R\mathbb{Z}$ an injective function on X.

Injectivity of d ensures that no two distinct elements of X have the same distance. We define a function on the set X called the baby-step, $\mathrm{bs} : X \to X$ as follows. Consider the following set

$$S_x = \{r \in \mathbb{R} \mid r > 0 \text{ and } d(x) + r \bmod R \in d(X)\}. \tag{1}$$

Let $f_x = \min S_x$. Then $\mathrm{bs}(x) = x'$ such that $d(x') = d(x) + f_x \mod R$. We also define the relative distance function

$$\Delta_{\mathrm{bs}} : X \to \mathbb{R} \text{ where } \Delta_{\mathrm{bs}}(x) = f_x = \min S_x. \tag{2}$$

Informally, the $\mathrm{bs}(x)$ gives the element next to x. The circumference of the infrastructure, denoted R, can be expressed in terms of this relative distance function as follows:

$$R = \sum_{i=0}^{m-1} \Delta_{\mathrm{bs}}(x_i). \tag{3}$$

It is clear that bs^{-1}, the inverse of bs, is well-defined. Further, a group-like structure is imposed on the set X by means of a binary operator, called the giant-step. Consider the set

$$S_{x,y} = \{r \in \mathbb{R} \mid r \geq 0 \text{ and } d(x) + d(y) + r \bmod R \in d(X)\}.$$

Let $f_{x,y} = \min S_{x,y}$. Then $\mathrm{gs} : X \times X \to X$ is defined as:

$$\mathrm{gs}(x, y) = z \text{ such that } d(z) = d(x) + d(y) + f_{x,y} \bmod R. \tag{4}$$

We define the relative distance function Δ_{gs} as:

$$\Delta_{\mathrm{gs}} : X \times X \to \mathbb{R} \text{ where } \Delta_{\mathrm{gs}}(x, y) = f_{x,y} = \min S_{x,y}. \tag{5}$$

The giant-step is commutative, but not associative. It is "almost associative" in the sense that for two arbitrary elements $x, y \in X$ the giant-step gives an element $z \in X$ whose distance satisfies $d(z) \approx d(x) + d(y)$.

In infrastructures arising out of quadratic number fields the elements of the infrastructure correspond to the principal reduced ideals of the number field. The distance function is

the norm of the ideals. One can cycle through these ideals using the so-called reduction operator [5]; this function corresponds to the baby-step. One can also define the product of ideals which after reduction corresponds to the giant-step, see [5].

The definitions of bs and gs and the relative distance functions Δ_{bs} and Δ_{gs} may suggest that we need R and the distance function d to be able to compute them. However, this is not the case. These functions can be computed efficiently without the knowledge of R or the distance function d. To illustrate this point, let us explain how (discrete) infrastructures can be considered as generalizations of finite cyclic groups.

Definition 2: (Discrete infrastructure) An infrastructure is said to be discrete if its circumference R is a positive integer and its distance function d is integer-valued, i.e., $d : X \hookrightarrow \mathbb{Z}/R\mathbb{Z}$.

Example 3: (Finite cyclic group) Suppose $G = \langle g \rangle$ is a finite cyclic group of order R and generated by g. Then we can form an infrastructure out of G as follows. We let $X = G$ and define $d(h) = \log_g h$, for any $h \in G$, since every element $h \in G$ is of the form $g^{d(h)}$ for some $d(h) \in \mathbb{Z}$. The baby step bs of the infrastructure corresponds simply to multiplication of elements x by the generator g, while the giant step gs corresponds to the multiplication of two elements x and y in G. The relative distance functions Δ_{bs} and Δ_{gs} are constant and take on the values 1 and 0, respectively.

We can now interpret the order of G as the circumference of the infrastructure. The distance function $d(x)$ corresponds to the discrete logarithm of the element x with respect to the base g. This example makes it clear why we cannot necessarily determine the circumference and the distance function efficiently, even though we can efficiently evaluate the baby and giant steps and their corresponding distance functions.

2.2 Computational problems

The main computational problems related to infrastructures are the computation of the circumference and the computation of generalized discrete logarithms.

We consider only infrastructures that satisfy the assumptions below. These are necessary to be able to carry out basic arithmetic operations in infrastructures in polynomial time. The cost is measured with respect to the input problem size n.

A1) The circumference satisfies $R \leq 2^{\text{poly}(n)}$.
A2) Any element $x \in X$ can be represented by a bit string of length poly(n).
A3) The elements bs(x), bs$^{-1}(x)$, gs(x, y) can be determined in time poly(n) for all $x, y \in X$.
A4) The relative distances $\Delta_{\text{bs}}(x)$ and $\Delta_{\text{gs}}(x, y)$ cannot necessarily be computed exactly. We only obtain approximate values $\tilde{\Delta}_{\text{bs}}(x)$ and $\tilde{\Delta}_{\text{bs}}(x, y)$ with

$$|\Delta_{\text{bs}}(x) - \tilde{\Delta}_{\text{bs}}(x)| < \frac{1}{2^m} \text{ and } |\Delta_{\text{gs}}(x, y) - \tilde{\Delta}_{\text{gs}}(x, y)| < \frac{1}{2^m} \tag{6}$$

in time[c] $\mathrm{poly}(n, m)$.

A5) The minimum distance $d_{\min}$ between any two elements of the infrastructure is bounded from below by

$$d_{\min} = \min_{x \in X} \{\Delta_{\mathrm{bs}}(x)\} \geq \frac{1}{2^{\mathrm{poly}(n)}}. \tag{7}$$

A6) The maximum distance $d_{\max}$ between any two elements of the infrastructure is bounded from above by

$$d_{\max} = \max_{x \in X} \{\Delta_{\mathrm{bs}}(x)\} \leq \mathrm{poly}(n). \tag{8}$$

A7) There exists a positive integer $\bar{k} \leq \mathrm{poly}(n)$ and a positive (rational) number $d_{\bar{k}} \geq \mathrm{poly}(n)$ such that for all $x \in X$ we have

$$\sum_{i=0}^{\bar{k}-1} \Delta_{\mathrm{bs}}(\mathrm{bs}^i(x)) \geq d_{\bar{k}}\,, \tag{9}$$

where bs^i denotes the i-fold application of bs. In words, any $\bar{k}$ consecutive elements span a distance of a least $d_{\bar{k}}$.

We emphasize that these assumptions are not restrictive; in fact, they are routinely made in the work on infrastructures. We have spelt them out explicitly for expository reasons. In particular, infrastructures arising from quadratic number fields satisfy all the assumptions made above; further justification for these assumptions for number fields is provided below. The first three assumptions are obvious. The relative distances Δ_{bs} and Δ_{gs} could be arbitrary real numbers and, thus, we cannot always obtain the exact values. Assumption **A**4 is made because we cannot perform arithmetic with arbitrary real numbers. Assumptions **A**5 – **A**7 ensure that we can compute in certain circle groups associated to infrastructures and evaluate certain homomorphisms into these groups efficiently in time $\mathrm{poly}(n)$.

The computational problems in infrastructures are :

- **Computation of the circumference**:
  determine an m-bit approximation of the circumference R
- **Generalized discrete logarithm problem**:
  given an element $y \in X$, determine an m-bit approximation of $d(y)$

The main contributions of this work are efficient quantum algorithms for infrastructures satisfying assumptions **A**1 – **A**7. These algorithms make it possible to determine $\lfloor R \rceil$ and $\lfloor d(y) \rceil$ in time $\mathrm{poly}(n)$, where the notation $\lfloor r \rceil$ means either the floor or ceiling of the real number r. Simple classical post processing allows us to obtain efficiently m-bit approximations from these integral approximations. For the sake of completeness, we prove later how this can be accomplished.

We now justify the validity of the above assumptions in the case of infrastructures from number fields of unit rank 1 (such number fields give rise to one-dimensional infrastructures).

[c]Note that m here and elsewhere in the rest of paper is not related to the number of elements in the infrastructure.

A1) This is shown in [17] (see also [18]).
A2) This is shown in [19, Corollary 3.7].
A3) In [12], it is shown that the baby steps and giant steps can be computed in $O(D^\epsilon)$ for arbitrary $\epsilon > 0$ (where D is the absolute value of the discriminate, which is bounded by $2^{\mathrm{poly}(n)}$). However, if one traces through their references and updates the analysis of the running time, one finds that everything is polynomial in $\log(D)$ and not just subexponential [20].
A4) This assumption is valid since one can approximate logarithms of absolute values of elements in number fields whose size is polynomially bounded in n.
A5) In [21, Example 9.4], it is shown that $d_{\min}$ can be of size $1/2^{\mathrm{poly}(n)}$. In [20], Fontein informed us that **A**5 holds in general.
A6) This is shown in [12, Proposition 2.7 (i)].
A7) This is shown in [12, Proposition 2.7 (ii)].

The infrastructures from function fields are always discrete. This means that there are no issues with finite precision. Therefore, the above computational problems can be solved directly with the standard hidden subgroup approach. This is because the circle groups corresponding to discrete infrastructures are just finite cyclic groups. In [20], Fontein informed us that the relevant assumptions also hold in infrastructures from finite fields.

2.3 Circle groups from infrastructures

We now show that infrastructures naturally give rise to circle groups that are isomorphic to $\mathbb{R}/R\mathbb{Z}$. This isomorphism is the key to solving the two computational problems in quantum polynomial time. Here and in the next two subsections, we assume that we can compute Δ_{bs} and Δ_{gs} exactly.

Picture the elements of X to be embedded in a circle of circumference R as follows. They are placed along the circle starting with x_0 at the topmost point of the circle and then moving clockwise. Their position is determined by the distance function d. For instance, the element x_i is associated to the point $d(x_i)$ on the circle as depicted in figure 1.

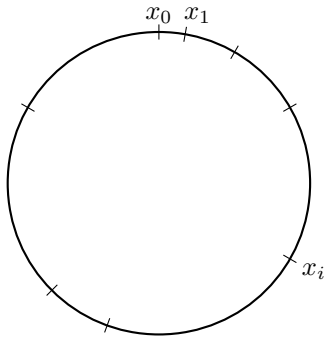

Fig. 1. Embedding an infrastructure into $\mathbb{R}/R\mathbb{Z}$

This embedding alone does not yet give rise to a valid group structure because $d(x_i) + d(x_j)$

is not necessarily an element of $d(X)$. To obtain a group, we start with the set $X \times \mathbb{R}$ and the map $\psi : X \times \mathbb{R} \to \mathbb{R}/R\mathbb{Z}$ defined by

$$\psi(x, f) = d(x) + f \tag{10}$$

for all $(x, f) \in X \times \mathbb{R}$. We call this the absolute distance of the pair (x, f).

For each $d \in \mathbb{R}/R\mathbb{Z}$, there exist infinitely many pairs $(x, f) \in X \times \mathbb{R}$ with $\psi(x, f) = d$. To avoid this infinitude, we continue by defining the equivalence relation $\equiv$ on $X \times \mathbb{R}$: two pairs $(x, f), (y, g) \in X \times \mathbb{R}$ are said to be equivalent if and only if $\psi(x, f) = \psi(y, g)$ (which is the same as $d(x)+f \equiv d(y)+g \bmod R$). We denote the equivalence class of (x, f) by $[x, f]$.

Now the set $X \times \mathbb{R}/\equiv$ can be endowed with a group structure as follows.

Proposition 4: The absolute distance map ψ in Eq. (10) is a group isomorphism from $\mathcal{G} := X \times \mathbb{R}/\equiv$ to $\mathbb{R}/R\mathbb{Z}$, where the (commutative) group operation on $\mathcal{G}$ is defined by

$$[x, f] + [y, g] := [\mathrm{gs}(x, y), f + g - \Delta_{\mathrm{gs}}(x, y)] \tag{11}$$

for arbitrary pairs $(x, f), (y, g) \in X \times \mathbb{R}$.

Proof: The proof is straightforward. We just verify that ψ is a group homomorphism. Letting $\psi(x, f) = d(x) + f$ and $\psi(y, g) = d(y) + g$, we obtain $\psi(\mathrm{gs}(x, y), f + g - \Delta_{\mathrm{gs}}(x, y)) = d(\mathrm{gs}(x, y)) + f + g - \Delta_{\mathrm{gs}}(x, y)$. By the definition of the giant-step it holds that $d(\mathrm{gs}(x, y)) = d(x) + d(y) + \Delta_{\mathrm{gs}}(x, y)$. Thus, $d(\mathrm{gs}(x, y)) + f + g - \Delta_{\mathrm{gs}}(x, y) = d(x) + d(y) + f + g = \psi(x, f) + \psi(y, g)$. $\square$.

2.4 Group arithmetic based on f-representations

We have to use "nice" representatives for the equivalence classes of $\mathcal{G}$ to be able to compute within this group efficiently. To this end, we introduce f-representations. Intuitively, the f-representations fill in the missing points in the circle $\mathbb{R}/R\mathbb{Z}$, i.e., the set of points $(\mathbb{R}/R\mathbb{Z}) \setminus d(X)$.

Definition 5: (f-representation) Let (X, d) be an infrastructure. A pair $(x, f) \in X \times \mathbb{R}$ is said to be an f-representation if $0 \leq f < \Delta_{\mathrm{bs}}(x)$. We denote the set of all f-representations by $\mathrm{Rep}(\mathcal{I})$.

The following lemma was shown in [14] (see Proposition 2 and Corollary 1 therein) in a slightly less general setting. We include this lemma for completeness. An important aspect of this lemma is that the group operation can be realized without having any knowledge of R or the distance function d (except for the knowledge that is revealed indirectly through the particular interplay of the functions bs, gs, Δ_{bs}, and Δ_{gs}).

We mention that for arbitrary infrastructures, neither this lemma nor any simple method make it possible to compute inverses in $\mathcal{G}$. However, in the case of infrastructures in global

fields there is an efficient classical way to compute (approximate) f-representations of inverses in the corresponding circle groups.

Lemma 6: The group operation in $\mathcal{G}$ can be efficiently realized by using f-representations to encode the equivalence classes. More precisely, it takes at most $\bar{k}\lceil 2d_{\max}/d_{\bar{k}}\rceil = \mathrm{poly}(n)$ invocations of baby steps to obtain the f-representation corresponding to the sum of two elements of $\mathcal{G}$.

Proof: Let $(x,f),(x',f') \in \mathrm{Rep}(\mathcal{I})$. Then, we have

$$[x,f]+[x',f'] = [\mathrm{gs}(x,x'), f+f'-\Delta_{\mathrm{gs}}(x,x')]\,.$$

In general, the pair $(x'',f'') := (\mathrm{gs}(x,x'), f+f'-\Delta_{\mathrm{gs}}(x,x')) \in X\times\mathbb{R}$ is not a valid f-representation. The task now is to find the f-representation that encodes the same equivalence class in $\mathcal{G}$ as (x'',f''). We use the bounds $-d_{\max} \le f'' = f+f'-\Delta_{\mathrm{gs}}(x,x') \le f+f' \le 2d_{\max}$, where $d_{\max}$ is the maximum distance between two consecutive elements of the infrastructure.

If $f'' \le 0$, then we iteratively replace (x'',f'') with $(\mathrm{bs}^{-1}(x''), f''+\Delta_{\mathrm{bs}}(x''))$ until it just becomes positive. If $f'' \ge 0$, then we iteratively replace (x'',f'') with $(\mathrm{bs}(x''), f''-\Delta_{\mathrm{bs}}(x''))$ until it is minimal while being nonnegative. Observe that this reduction process preserves the absolute distance. Moreover, it takes at most $\bar{k}\lceil 2d_{\max}/d_{\bar{k}}\rceil = \mathrm{poly}(n)$ steps to obtain the canonical representative in $\mathrm{Rep}(\mathcal{I})$ $\square$.

From now on, we identify $\mathcal{G}$ with $\mathrm{Rep}(\mathcal{I})$ and use $(x,f) \in \mathrm{Rep}(\mathcal{I})$ to denote the group elements instead of $[x,f]$ to simplify notation.

The corollary below is a simple consequence of the above lemma. We state it explicitly because this result is extensively used in the quantum algorithms.

Corollary 7: (Double & multiply) Let $(x,f) \in \mathcal{G}$ be an arbitrary group element and $a \in \mathbb{Z}$ an arbitrary nonnegative integer. Then, it takes at most $O(\bar{k}\lceil 2d_{\max}/d_{\bar{k}}\rceil \log(a)) = \mathrm{poly}(n)\log(a)$ invocations of baby steps and at most $O(\log(a))$ invocations of giant steps to obtain the f-representation corresponding to $a\cdot(x,f)$.

Proof: The action of $\mathbb{Z}$ on the commutative group $\mathcal{G}$ is defined by

$$a\cdot(x,f) \quad := \quad \underbrace{(x,f)+(x,f)+\cdots+(x,f)}_{a \text{ times}}\,.$$

Consider the special case of computing $a\cdot(x,f)$ for $a=2^i$ with some i. This takes at most

$O(i)$ steps:

$$\begin{aligned}
(x,f) &= (x^{(0)},f^{(0)}) \\
2(x,f) &= (x^{(0)},f^{(0)})+(x^{(0)},f^{(0)}) = \big(\mathrm{gs}(x^{(0)},x^{(0)}), 2f^{(0)}-\Delta_{\mathrm{gs}}(x^{(0)},x^{(0)})\big) \\
&=: (x^{(1)},f^{(1)}) \\
&\vdots \\
2^i(x,f) &= (x^{(i-1)},f^{(i-1)})+(x^{(i-1)},f^{(i-1)}) \\
&= \big(\mathrm{gs}(x^{(i-1)},x^{(i-1)}), 2f^{(i-1)}-\Delta_{\mathrm{gs}}(x^{(i-1)},x^{(i-1)})\big) \qquad (12)\\
&=: (x^{(i)},f^{(i)})\,.
\end{aligned}$$

In each step, we apply the above lemma to ensure that $(x^{(i)},f^{(i)})$ are valid f-representations.

Now suppose $a = b_i 2^i + b_{i-1}2^{i-1} + \cdots + b_0 2^0$ in binary representation. Then, $a\cdot(x,f)$ can be computed as

$$a\cdot(x,f) = \sum_{j=0}^{i} b_j \cdot (x^{(j)},f^{(j)})\,.$$

with at most i additions. We again use the above lemma to ensure that the partial sums are valid f-representations.

In total, the whole process takes at most $O(\log(a))$ giant-steps and $O(\bar{k}\lceil 2d_{\max}/d_{\bar{k}}\rceil \log(a))$ baby-steps $\square$.

2.5 Group homomorphisms from $\mathbb{R}$ and $\mathbb{Z}\times\mathbb{R}$ into circle groups

In this subsection, we continue to assume that we can determine the functions Δ_{bs} and Δ_{gs} exactly, and compute with arbitrary real numbers. In the next subsection, we will relax this assumption.

Definition 8: Let $h:\mathbb{R}\to\mathcal{G}$ be the surjective group homomorphism, where $h(r)$ is defined to be the unique f-representation $(x,f)\in \mathrm{Rep}(\mathcal{I})$ with $(x_0,r)\equiv(x,f)$.

Recall that we define the distance function d such that $d(x_0)=0$, thus $(x_0,0)$ is the identity element of $\mathcal{G}$.

The statement of the following lemma is obvious. We formulate it explicitly since it provides the intuition required to understand the quantum algorithm for computing the circumference.

Lemma 9: The kernel of h is equal to $R\mathbb{Z}$. Thus, h is a periodic function on $\mathbb{R}$ with period R.

Lemma 10: Let $r\in[0,B]\subset\mathbb{R}$, where B is an arbitrary (but fixed) positive real number. Then, we can determine the exact value $h(r)$ using $O(\log(B)\bar{k}\lceil 2d_{\max}/d_{\bar{k}}\rceil) = O(\log(B)\mathrm{poly}(n))$

baby-steps and $O(\log(B))$ giant-steps under the assumption that $\Delta_{\rm bs}$ and $\Delta_{\rm gs}$ can be computed exactly.

Proof: In general, (x_0, r) is not a valid f-representation. Thus, we need to find the corresponding f-representation. If r is small and positive, then we can use baby-steps to find it with at most $\bar{k}\lceil r/d_{\bar{k}}\rceil$ invocations.

If r is large, then the baby-step method is not efficient anymore. We have to use giant-steps as well. The idea is to use the double and multiply technique of Corollary 7. Let $x_{\bar{k}} = {\rm bs}^{\bar{k}}(x_0)$. Then $d(x_{\bar{k}}) \geq d_{\bar{k}}$. Let $a = [r/d(x_{\bar{k}})]$, where $[x]$ denotes the nearest integer to x. We can compute $a \cdot (x_{\bar{k}}, 0) = (x, f)$ using $O(\log(a)) = O(\log(B))$ giant-steps and $O(\log(B)\bar{k}\lceil 2d_{\max}/d_{\bar{k}}\rceil)$ baby-steps. Note that $(x, f) \equiv (x_0, ad(x_{\bar{k}}))$. But, $|ad(x_{\bar{k}}) - r| = |[r/d(x_{\bar{k}})]d(x_{\bar{k}}) - r| \leq d(x_{\bar{k}})/2$. Therefore, (x, f) is at most within a distance of $d(x_{\bar{k}})/2$ from r. Thus we can find $h(r)$ by using no more than $\bar{k}$ additional invocations of either bs or ${\rm bs}^{-1}$. The overall time complexity of evaluating $h(r)$ is therefore $O(\log(B)\bar{k}\lceil 2d_{\max}/d_{\bar{k}}\rceil) = O(\log(B){\rm poly}(n))$, since $d_{\max}$ and $\bar{k}$ are $O({\rm poly}(n)$ by assumptions **A**6 and **A**7 □.

Similar ideas can be applied when r is negative. The method proposed in Lemma 10 relies essentially on the group arithmetic of $\mathcal{G}$ and thus is quite different from a generalization of the binary search method.

Definition 11: Let $x \in X$ be an arbitrary (but fixed) element of the infrastructure. Let $g : \mathbb{Z} \times \mathbb{R} \to \mathcal{G}$ be the surjective homomorphism, where $g(a, r)$ is defined to be the unique f-representation corresponding to

$$a \cdot (x, 0) + h(r)\,. \tag{13}$$

We note that $g(a, b)$ is same as the f-representation of $h(ad(x)+b)$, where $d(x)$ is the distance of x.

The following statement on the kernel of the homomorphism g is obvious.

Lemma 12: The kernel of the above homomorphism g is equal to

$$\{(a, r) \,:\, r \equiv -a\, d(x) \mod R\}\,.$$

Corollary 13: Let A be an arbitrary positive integer and B an arbitrary positive real number. Then, we can determine the exact value $g(a, b)$ for all pairs $(a, r) \in \{0, 1, \ldots, A-1\} \times [0, B]$ in time $O((\log A + \log B){\rm poly}(n))$ under the assumption that $\Delta_{\rm bs}$ and $\Delta_{\rm gs}$ can be computed perfectly.

Proof: By definition $g(a, r) = a \cdot (x, 0) + h(r)$. The computation of $a \cdot (x, 0)$ can be performed in $O(\log(A)\bar{k}\lceil 2d_{\max}/d_{\bar{k}}\rceil) = O(\log(A){\rm poly}(n))$ time by Corollary 7, while the computation of $h(r)$ can be performed in $O(\log(B)\bar{k}\lceil 2d_{\max}/d_{\bar{k}}\rceil) = O(\log(B){\rm poly}(n))$ time by Lemma 10. The final group addition in $\mathcal{G}$ takes at most $\bar{k} = {\rm poly}(n)$ baby-steps, by Lemma 6 □.

2.6 Efficient approximate group arithmetic and evaluation of the homomorphisms from $\mathbb{R}$ and $\mathbb{Z} \times \mathbb{R}$

The previous assumption that we can compute $\Delta_{\rm bs}$ and $\Delta_{\rm gs}$ and represent arbitrary real numbers is clearly an idealization. We made this assumption at first because we can explain the intuition in a simpler and more elegant way when the homomorphisms h and g are perfect. We now drop this assumption and work instead with the approximate versions $\tilde{\Delta}_{\rm bs}$ and $\tilde{\Delta}_{\rm gs}$.

Let L be some large positive integer. We only consider evaluation points r that are rational numbers with denominator L.

Let $h(r) = (x, f)$ be the perfect f-representation with $(x, f) \equiv (x_0, r)$. We can only determine an approximate $\tilde{h}(r) = (\tilde{x}, \tilde{f}) \in X \times \mathbb{R}$ of $h(r)$. This approximation can be realized efficiently and has the following two properties:

P1. The first component is off at most by either a baby-step backward or forward, i.e., $\tilde{x} \in \{{\rm bs}^{-1}(x), x, {\rm bs}(x)\}$.

P2. If we have the promise that

$$\frac{1}{L} \leq f \leq \Delta_{\rm bs}(x) - \frac{1}{L} \tag{14}$$

holds, then the first component is correct, i.e., $\tilde{x} = x$, and the second component $\tilde{f}$ satisfies

$$|f - \tilde{f}| \leq \frac{1}{2L}\,. \tag{15}$$

Later, we will show that all evaluation points r necessary for the quantum algorithm are such that the condition in Eq. (14) holds with high probability by adding a random shift to the evaluation points.

Lemma 14: Let L be a positive integer with $d_{\rm min} > 1/L$. We consider only evaluation points of the form $r = k/L$ with $r < B$. Let $h(r) = (x, f)$ be the perfect f-representation. Then, we can compute an approximate pair $\tilde{h}(r) = (\tilde{x}, \tilde{f})$ that satisfies **P1**, **P2**. The running time is ${\rm poly}(\log(B), \log(L), n)$.

Proof: We analyze what happens if we run the algorithm in Lemma 10, but now rely on the approximate versions $\tilde{\Delta}_{\rm bs}$ and $\tilde{\Delta}_{\rm gs}$. Recall that the parameter m characterizes the precision of the approximations. The maximal deviation between the approximate and perfect values is smaller than $1/2^m$.

We use $\tilde{d}_{\rm acc}(\cdot)$ to denote the corresponding approximate accumulated distances of the (intermediate) f-representations and their first components. We use $d_{\rm acc}(\cdot)$ to denote the correct accumulated distance of the representations and elements (these distances exist even though we cannot always compute them). The accumulated distances are not taken modulo R and take into account how the f-representation is generated. A key observation that

we need in the proof is that $d_{\mathrm{acc}}(\tilde{x},\tilde{f}) = d_{\mathrm{acc}}(\tilde{x}) + \tilde{f}$ and $\tilde{d}_{\mathrm{acc}}(\tilde{x},\tilde{f}) = \tilde{d}_{\mathrm{acc}}(\tilde{x}) + \tilde{f}$, so that $d_{\mathrm{acc}}(\tilde{x},\tilde{f}) - \tilde{d}_{\mathrm{acc}}(\tilde{x},\tilde{f}) = d_{\mathrm{acc}}(\tilde{x}) - \tilde{d}_{\mathrm{acc}}(\tilde{x})$.

The characterizing condition of the perfect f-representation is

$$d_{\mathrm{acc}}(x) \le r < d_{\mathrm{acc}}(x) + \Delta_{\mathrm{bs}}(x)\,. \tag{16}$$

We can only guarantee

$$\tilde{d}_{\mathrm{acc}}(\tilde{x}) \le r < \tilde{d}_{\mathrm{acc}}(\tilde{x}) + \tilde{\Delta}_{\mathrm{bs}}(\tilde{x}) \tag{17}$$

for the approximate pair $(\tilde{x},\tilde{f})$.

Assume that m has been chosen to be sufficiently large so that

$$|\tilde{d}_{\mathrm{acc}}(\tilde{x}) - d_{\mathrm{acc}}(\tilde{x})| \le \frac{1}{2L} \tag{18}$$

holds. Together with Eq. (17) this implies

$$d_{\mathrm{acc}}(\tilde{x}) - \frac{1}{2L} \le r < d_{\mathrm{acc}}(\tilde{x}) + \Delta_{\mathrm{bs}}(\tilde{x}) + \frac{1}{2L} + \frac{1}{2^m}\,. \tag{19}$$

This condition on $\tilde{x}$ is weaker than the condition of the perfect x in Eq. (16). But since $1/2^m < 1/L < d_{\min}$ we must have $\tilde{x} \in \{\mathrm{bs}^{-1}(x), x, \mathrm{bs}\,(x)\}$, depending on which of the three cases $r < d_{\mathrm{acc}}(\tilde{x})$, $d_{\mathrm{acc}}(\tilde{x}) \le r < \tilde{d}_{\mathrm{acc}}(\tilde{x}) + \Delta_{\mathrm{bs}}(\tilde{x})$, or $\tilde{d}_{\mathrm{acc}}(\tilde{x}) + \Delta_{\mathrm{bs}}(\tilde{x}) \le r$ occurs. We cannot have a deviation by more than one baby-step backward or forward because otherwise Eq. (17) would not be satisfied.

If we know that f satisfies $\frac{1}{L} \le f \le \Delta_{\mathrm{bs}}(x) - \frac{1}{L}$, then we can conclude that $\tilde{x} = x$ must hold. This is because the first and third cases are excluded. The condition on $\tilde{f}$ is automatically satisfied in this case since $\tilde{f} = r - \tilde{d}_{\mathrm{acc}}(\tilde{x})$, which is the same as $r - \tilde{d}_{\mathrm{acc}}(x)$.

We now show how to choose m so that the condition in Eq. (17) holds. The algorithm in Lemma 10 has two steps. In the first step, we compute $a \cdot (x_{\bar{k}}, 0)$, where $a = [r/d(x_{\bar{k}})]$. This gives us a representation (x', f'), such that

$$|d_{\mathrm{acc}}(x', f') - r| \le \frac{d(x_{\bar{k}})}{2}.$$

Then we apply a sequence of baby-steps to obtain an f-representation (x, f), which satisfies $d_{\mathrm{acc}}(x, f) = r$.

Working with $\tilde{\Delta}_{\mathrm{bs}}$ and $\tilde{\Delta}_{\mathrm{gs}}$, in the first step we actually compute $(\tilde{x}', \tilde{f}')$ an approximation of $\tilde{a} \cdot (x_{\bar{k}}, 0)$, where $\tilde{a} = [r/\tilde{d}(x_{\bar{k}})]$.

Let us analyze the error in this computation. The computation of $\tilde{a} \cdot (x_{\bar{k}}, 0)$ itself can be broken down into two parts: (i) computation of representations of the form $(\tilde{x}^{(i)}, \tilde{f}^{(i)})$ which approximate $2^i(x_{\bar{k}}, 0)$ and (ii) summing $O(\log \tilde{a})$ such representations.

The error at the very beginning e_0 satisfies

$$e_0 := |\tilde{d}_{\mathrm{acc}}(\tilde{x}^{(0)}) - d_{\mathrm{acc}}(\tilde{x}^{(0)})| = |\tilde{d}_{\mathrm{acc}}(\tilde{x}^{(0)}, \tilde{f}^{(0)}) - d_{\mathrm{acc}}(\tilde{x}^{(0)}, \tilde{f}^{(0)})| < \frac{\bar{k}}{2^m}\,.$$

Note that $\tilde{d}_{\mathrm{acc}}(\tilde{x}^{(0)}) \geq d_{\bar{k}}$ holds because if we get a value strictly smaller than $d_{\bar{k}}$ we can replace it by $d_{\bar{k}}$, because of **A**7. The error in the ith step

$$e_i := |\tilde{d}_{\mathrm{acc}}(\tilde{x}^{(i)}) - d_{\mathrm{acc}}(\tilde{x}^{(i)})| = |\tilde{d}_{\mathrm{acc}}(\tilde{x}^{(i)}, \tilde{f}^{(i)}) - d_{\mathrm{acc}}(\tilde{x}^{(i)}, \tilde{f}^{(i)})|$$

satisfies the recursion

$$e_i < 2e_{i-1} + \frac{1}{2^m} + \frac{\bar{k}}{2^m}\left\lceil \frac{2d_{\max}}{d_{\bar{k}}} \right\rceil \tag{20}$$

The recursion relation can be easily explained by considering Eq. (12). The first term is due to the fact that the error in $\tilde{f}^{(i-1)}$ is multiplied by 2, the second term is due to one giant-step, and the third term due to $O(\bar{k}\lceil 2d_{\max}/d_{\bar{k}}\rceil)$ baby-steps used to obtain a valid f-representation. This implies

$$e_i < \frac{1}{2^{m-i}}\left(\bar{k} + 1 + \bar{k}\left\lceil \frac{2d_{\max}}{d_{\bar{k}}} \right\rceil\right). \tag{21}$$

In order to obtain $(\tilde{x}', \tilde{f}')$, we have to sum $O(\log \tilde{a})$ such f-representations, where i varies from 0 to $\log \tilde{a} - 1$. Each sum adds an additional error term due to the giant step and the baby-steps used for reduction. Therefore the error at the end of the first step is given by

$$\begin{aligned} e' &:= |\tilde{d}_{\mathrm{acc}}(\tilde{x}') - d_{\mathrm{acc}}(\tilde{x}')| = |\tilde{d}_{\mathrm{acc}}(\tilde{x}', \tilde{f}') - d_{\mathrm{acc}}(\tilde{x}', \tilde{f}')| \\ &= \frac{\tilde{a}}{2^m}\left(\bar{k} + 1 + \bar{k}\left\lceil \frac{2d_{\max}}{d_{\bar{k}}} \right\rceil\right) + \frac{\log \tilde{a}}{2^m}\left(1 + \bar{k}\left\lceil \frac{2d_{\max}}{d_{\bar{k}}} \right\rceil\right), \\ &\leq \frac{r + d_{\bar{k}}}{2^m d_{\bar{k}}}\left(\bar{k} + 1 + \bar{k}\left\lceil \frac{2d_{\max}}{d_{\bar{k}}} \right\rceil\right) + \frac{\log(r/d_{\bar{k}} + 1)}{2^m}\left(1 + \bar{k}\left\lceil \frac{2d_{\max}}{d_{\bar{k}}} \right\rceil\right) \end{aligned}$$

where we used the fact that $\tilde{a} \leq r/d_{\bar{k}} + 1$. The f-representation (x', f') is at most at a distance[d] of $d_{\max}\bar{k}$ from r. Thus $(\tilde{x}', \tilde{f}')$ is at most a distance of $(e' + d_{\max}\bar{k})$ from r and we need to take at most $\bar{k}\lceil (e' + d_{\max}\bar{k})/d_{\bar{k}} \rceil$ baby-steps to obtain $(\tilde{x}, \tilde{f})$.

The error in the accumulated distances of the final representation $(\tilde{x}, \tilde{f})$ is given by

$$\begin{aligned} \tilde{e} &:= |\tilde{d}_{\mathrm{acc}}(\tilde{x}) - d_{\mathrm{acc}}(\tilde{x})| = |\tilde{d}_{\mathrm{acc}}(\tilde{x}, \tilde{f}) - d_{\mathrm{acc}}(\tilde{x}, \tilde{f})| \\ &= e' + \frac{\bar{k}}{2^m}\left\lceil \frac{e' + d_{\max}\bar{k}}{d_{\bar{k}}} \right\rceil \end{aligned}$$

The dominant term in the error is the first term e', as it is proportional to r, while the second term is proportional to $r/2^m$ and therefore does not contribute too much as m is large. We can make the error smaller than $1/2L$ as required in Eq. (18) by choosing $m = \mathrm{poly}(\log(B), \log(L))$ □.

The proof does not actually require that the evaluation points are of the form k/L. Analogous results hold for the homomorphism g. We state them without proof since the above argument can be easily adapted.

Let $g(a, r) = (x, f)$ be the perfect f-representation with $(x, f) \equiv (x_0, r)$. We can only determine an approximate $\tilde{g}(a, r) = (\tilde{x}, \tilde{f}) \in X \times \mathbb{R}$ of $g(a, r)$. This approximation can be realized

[d] We can tighten this by a factor of 2. But this suffices.

efficiently and has the properties **P1**, **P2**.

Lemma 15: Let L be a positive integer with $d_{\min} > 1/L$. We consider only evaluation points of the form (a, r) with $a \in \{0, 1, \ldots, A-1\}$ and $r = k/L \in [0, B]$. Let $g(a, r) = (x, f)$ be the perfect f-representation. Then, we can compute an approximate pair $\tilde{g}(a, r) = (\tilde{x}, \tilde{f})$ that satisfies **P1**, and **P2**. The running time is $\mathrm{poly}(\log(A), \log(B), \log(L), n)$.

3 Quantum Algorithm for Approximating the Period of Pseudo-periodic States

In this section we generalize the notion of periodic states introduced in [22]. We assume that the quantum states are elements of a q-dimensional complex Hilbert space, denoted by $\mathbb{C}^q$.

3.1 Pseudo-periodic states

Definition 16: (Periodic state) A quantum state in $\mathbb{C}^q$ is periodic with period $r \in \mathbb{Z}$ at offset $k \in \{0, 1, \ldots, r-1\}$ if it is of the form

$$|\psi\rangle_{k,r} := \frac{1}{\sqrt{p}} \sum_{j=0}^{p-1} |k + jr\rangle, \tag{22}$$

where $p = \lfloor (q - k - 1)/r + 1 \rfloor$. We denote a periodic state with period r at offset k by $|\psi\rangle_{k,r}$.

Periodic states can be created by the evaluation of injective functions over a uniform superposition. To be more precise, we create the state $|\psi\rangle = q^{-1/2} \sum_{i=0}^{q-1} |i\rangle|f(i)\rangle$, and measure the second register. We assume that f is periodic with period r. It is possible to recover the period r by means of Fourier sampling. In fact, the period can be recovered even when r is irrational. For this reason, we generalize these periodic states to a larger class of quantum states called the pseudo-periodic states.

Definition 17: (Pseudo-periodic state) A pseudo-periodic state in $\mathbb{C}^q$, with possibly irrational period $r \in \mathbb{R}$, is of the form:

$$|\psi\rangle_{k,r} \quad = \quad \frac{1}{\sqrt{p}} \sum_{j=0}^{p-1} |\lfloor k + jr \rceil\rangle, \tag{23}$$

where $k \in \{0, 1, \ldots, \lfloor r \rfloor\}$ and p is the largest integer such that $\lfloor k + (p-1)r \rceil \leq (q-1)$.

Note that $\lfloor x \rceil$ can be either $\lfloor x \rfloor$ or $\lceil x \rceil$ so that $p \in \{\lfloor (q-2)/r \rfloor, \ldots, \lfloor q/r \rfloor + 1\}$, depending on the particular value of the offset k. If we assume that $r > 2$, then we can restrict $p \in \{\lfloor q/r \rfloor - 1, \lfloor q/r \rfloor, \lfloor q/r \rfloor + 1\}$.

The weakly periodic functions defined in [4] are one class of functions which can induce such pseudo-periodic states. As we show in this section, we can recover the period even when the state is "almost" periodic. We observe that in the definition of the periodic states above,

there is an implicit dependence on the offset k; this offset is usually the outcome of some measurement, and therefore random.

3.2 Perturbed geometric sums with missing terms

The following lemma is at the heart of the analysis of the quantum algorithms for infrastructures. It is crucial for understanding the performance of these algorithms. The special case $\mathcal{J} = \{0, 1, \ldots, n-1\}$ suffices to bound the probability of the algorithm for computing the circumference. The more general case where $\mathcal{J}$ is a proper subset of $\{0, 1, \ldots, n-1\}$ is necessary for the analysis of the quantum algorithm for computing the discrete logarithms.

Lemma 18: (Perturbed geometric sums with missing terms) Let ω be the nth root of unity $e^{2\pi i/n}$, $n \geq 2$, θ an arbitrary real-valued function defined on $\mathcal{J} \subseteq \{0, \ldots, n-1\}$ satisfying the following conditions on θ_j and $|\mathcal{J}|$:

$$|\theta_j| \leq n/32 \tag{24a}$$

$$|\mathcal{J}| \geq n(1 - c_\delta)/(1 - 2\sin(\pi/32)) \tag{24b}$$

where

$$c_\delta \quad = \quad \operatorname{sinc}(\delta) = \frac{\sin(\pi\delta)}{\pi\delta} \qquad \text{if } |\delta| < 1. \tag{25}$$

Then the following inequality holds:

$$\frac{1}{|\mathcal{J}|^2}\left|\sum_{j\in\mathcal{J}} \omega^{\delta j+\theta_j}\right|^2 \quad \geq \quad \left(1 - 2\sin(\pi/32) - (1 - c_\delta)\frac{n}{|\mathcal{J}|}\right)^2. \tag{26}$$

Proof: Triangle inequality and upper bound on the absolute value of the unperturbed geometric sum without missing terms imply

$$\begin{aligned}
\left|\sum_{j\in\mathcal{J}} \omega^{\delta j}\right| + \left|\sum_{j\in\bar{\mathcal{J}}} \omega^{\delta j}\right| \quad &\geq \quad \left|\sum_{j\in\mathcal{J}} \omega^{\delta j} + \sum_{j\in\bar{\mathcal{J}}} \omega^{\delta j}\right| \\
&= \quad \left|\sum_{j=0}^{n-1} \omega^{\delta j}\right| = \left|\frac{1-\omega^{\delta n}}{1-\omega^{\delta}}\right| \\
&= \quad \left|\frac{\sin(\pi\delta)}{\sin(\pi\delta/n)}\right| \qquad (27) \\
&\geq \quad \left|\frac{\sin(\pi\delta)}{\pi\delta/n}\right| = nc_\delta, \qquad (28)
\end{aligned}$$

The equality in Eq. (27) follows from $|1 - e^{i\vartheta}| = |e^{-i\vartheta/2} - e^{i\vartheta/2}| = 2|\sin(\vartheta/2)|$ holding for all $\vartheta \in \mathbb{R}$. For the inequality in Eq. (28) we used the fact that $|\sin\vartheta| \leq |\vartheta|$, when $|\vartheta| < \pi/2$.

Subtracting the absolute value of the sum over $\bar{\mathcal{J}}$ from both sides of Eq. (28) and dividing by $|\mathcal{J}|$ yields

$$\begin{aligned}\frac{1}{|\mathcal{J}|}\left|\sum_{j\in\mathcal{J}}\omega^{\delta j}\right| &\geq c_\delta\frac{n}{|\mathcal{J}|}-\frac{1}{|\mathcal{J}|}\left|\sum_{j\in\bar{\mathcal{J}}}\omega^{\delta j}\right| \\ &\geq c_\delta\frac{n}{|\mathcal{J}|}-\frac{\bar{\mathcal{J}}}{|\mathcal{J}|} \\ &= 1-\Big(1-c_\delta\Big)\frac{n}{|\mathcal{J}|}\,. \end{aligned} \tag{29}$$

We now bound the "perturbed" geometric sum. To this end, we use some basic ideas from quantum information theory. Define the states $|\psi\rangle = \frac{1}{\sqrt{|\mathcal{J}|}}\sum_{j\in\mathcal{J}}\omega^{\delta j}|j\rangle$ and $|e\rangle = \frac{1}{\sqrt{|\mathcal{J}|}}\sum_{j\in\mathcal{J}}|j\rangle$, the projector $P=|e\rangle\langle e|$, and the diagonal unitary matrix $U=\mathrm{diag}(\omega^{\theta_0},\ldots,\omega^{\theta_{|\mathcal{J}|-1}})$. Observe that the square of the absolute value of unperturbed geometric sum is equal to $\|P|\psi\rangle\|^2$ and that of the perturbed one to $\|PU|\psi\rangle\|^2$. We have

$$\begin{aligned}\Big|\|P|\psi\rangle\|-\|PU|\psi\rangle\|\Big| &\leq \|P|\psi\rangle-PU|\psi\rangle\| \\ &\leq \|P\|\cdot\|I-U\|\cdot\||\psi\rangle\| \\ &= 2\max_j\Big\{|\sin(2\pi\theta_j/(2n))|\Big\} \\ &\leq 2\sin(\pi/32)\,. \end{aligned} \tag{30}$$

The upper bound on $\|I-U\|$ follows by noting that the entries of the diagonal matrix $I-U$ are $1-e^{2\pi i\theta_j/n}$ and using the above identity for the absolute value of expressions of this form. Let $\|P|\psi\rangle\|=x$ and $\|PU|\psi\rangle\|=y$. Then Eq. (30) implies the desired result since

$$y^2\geq\big(x-2\sin(\pi/32)\big)^2\geq\left(1-2\sin(\pi/32)-(1-c_\delta)\,\frac{n}{|\mathcal{J}|}\right)^2 \tag{31}$$

where we used Eq. (29) in the last step $\square$.

We pause to make two observations regarding the application of this result. First, we must ensure that $|\mathcal{J}|/n\geq(1-c_\delta)/(1-2\sin(\pi/32))$ for $\delta\in[0,1)$. Second, the choice of $|\theta_j|\leq n/32$, can be improved in that we can tolerate a higher perturbation, depending on the actual value of δ. Although, we retain this bound on θ_j throughout this paper for the sake of a clearer exposition, optimizing this bound on θ_j based on δ will enable us to obtain better bounds on the success probability of the quantum algorithms.

3.3 Presentation and proof of the quantum algorithm

Now we shall give a quantum algorithm for estimating the period of a pseudo-periodic state. In general, these states arise from some periodic functions, therefore the proposed quantum algorithm can be used to estimate the periods of such functions.

Theorem 19: Given a pair of pseudo-periodic states whose period $S\in\mathbb{R}$ is bounded as $M\geq S>1$, then with a probability $\Omega(1)$ and in time $\mathrm{poly}(\log S)$, Algorithm 1 gives a list of

real numbers $\mathcal{L}$ such that for some $\hat{S} \in \mathcal{L}$, we have $|S - \hat{S}| \leq 1$. Further, $|\mathcal{L}| = O(\text{poly} \log S)$ and the success probability is given by

$$p_{\text{success}} \geq \frac{1}{2}\left(\frac{1}{32} - \frac{2}{S}\right)^2 \left(1 - \frac{2S}{q}\right)^2 \left(\text{sinc}\left(\frac{1}{2} + \frac{1}{2S}\right) - 2\sin(\pi/32)\right)^4 \tag{32}$$

where $M^2 \leq q < 2M^2$.

Algorithm 1 Approximate period of pseudo-periodic states

Require: A pair of pseudo-periodic states in $\mathbb{C}^q$ with period $S \in \mathbb{R}$, where M is an upper bound on $S > 2$ and q is an integer such that $S^2 \leq M^2 \leq q < 2M^2$.

1: For each pseudo-periodic state, apply a Fourier transform over $\mathbb{Z}_q$ and measure to obtain c and d.
2: Compute the convergents c_i/d_i of c/d where $d_i \leq \lfloor q/32 \rfloor$.
3: Return $\mathcal{L} = \left\{ [c_i q/c] \mid d_i \leq \lfloor q/32 \rfloor \right\}$ as candidates for S.

Proof: Assume that the pseudo-periodic state is as follows:

$$|\psi\rangle_{o,S} = \frac{1}{\sqrt{|\mathcal{J}|}} \sum_{j \in \mathcal{J}} |\lfloor o + jS \rceil\rangle \,.$$

where $\mathcal{J} = \{0, 1, \ldots, p-1\}$ and $p \in \{\lfloor q/S \rfloor - 1, \lfloor q/S \rfloor, \lfloor q/S \rfloor + 1\}$. Since we are Fourier sampling, we may assume without loss of generality, that $o = 0$. Therefore, the measured distribution will be the same as the one induced by Fourier sampling the following state:

$$\frac{1}{\sqrt{|\mathcal{J}|}} \sum_{j \in \mathcal{J}} |\lfloor jS \rceil\rangle \tag{33}$$

Taking the Fourier transform over $\mathbb{Z}_q$ we obtain

$$\frac{1}{\sqrt{|\mathcal{J}|}} \frac{1}{\sqrt{q}} \sum_{j \in \mathcal{J}} \sum_{\ell=0}^{q-1} \omega_q^{\ell \lfloor jS \rceil} |\ell\rangle. \tag{34}$$

The Fourier transform at $|\ell\rangle$ has the amplitude

$$\frac{1}{\sqrt{q|\mathcal{J}|}} \sum_{j \in \mathcal{J}} \omega_q^{\lfloor jS \rceil \ell} \,. \tag{35}$$

We seek to find a lower bound on the probability of obtaining outcomes ℓ of the form $[\frac{mq}{S}]$, where $m \in \{0, 1, \ldots, \lfloor S \rfloor\}$. For a given m, $[m\frac{q}{S}]$ denotes either the floor or ceiling so that that $m\frac{q}{S} = [m\frac{q}{S}] + \epsilon_\ell$ with $|\epsilon_\ell| \leq \frac{1}{2}$. The probability of observing ℓ is given by

$$\frac{1}{q|\mathcal{J}|} \left| \sum_{j \in \mathcal{J}} \omega_p^{\frac{p}{q} \lfloor jS \rceil [m\frac{q}{S}]} \right|^2 \,. \tag{36}$$

To bound this probability, we consider the exponent of ω_p

$$\begin{aligned}\frac{p}{q}\lfloor jS\rceil[m\frac{q}{S}] &= \frac{p}{q}(jS+\delta_j)(m\frac{q}{S}+\epsilon_\ell)\\ &= pmj+\frac{pS\epsilon_\ell}{q}j+\frac{p\delta_j\epsilon_\ell}{q}+\frac{pm\delta_j}{S}\\ &= pmj+\frac{pS\epsilon_\ell}{q}j+\frac{p\delta_j\epsilon_\ell}{q}+\frac{pm\delta_j}{S}\\ &= pmj+\delta j+\theta_j. \end{aligned} \tag{37}$$

The first term is a multiple of p, implying that it can be omitted in the exponent. The factor $\delta = \frac{pS\epsilon_\ell}{q}$ in front of j in the second term is less or equal to $(1 + S/q)/2 \leq (1 + 1/S)/2$. The absolute value of the sum of the third and fourth terms is less or equal to $p/32$ provided that $m < \lfloor S/32 \rfloor$. In this case, the phase perturbations θ_j caused by these two terms satisfy Eq. (24a). Further, $|\mathcal{J}| = p$ ensures that Eq. (24b) is also satisfied and we can apply Lemma 18. We conclude that the probability of obtaining $|\ell\rangle$ is

$$\begin{aligned}\frac{1}{q|\mathcal{J}|}\left|\sum_{j\in\mathcal{J}}\omega_p^{\frac{p}{q}\lfloor jS\rceil[m\frac{q}{S}]}\right|^2 &\geq \frac{p}{q}\left(\operatorname{sinc}\Big(\frac{1}{2}+\frac{1}{2S}\Big)-2\sin(\pi/32)\right)^2\\ &\geq \left(\frac{1}{S}-\frac{2}{q}\right)\left(\operatorname{sinc}\Big(\frac{1}{2}+\frac{1}{2S}\Big)-2\sin(\pi/32)\right)^2\end{aligned}$$

where the last inequality follows from $p \geq \lfloor q/S \rfloor - 1 \geq q/S - 2$. So the probability of obtaining any "good" ℓ, i.e. $m \in \{1, \ldots, \lfloor S/32 - 1 \rfloor\}$, is at least β, where

$$\beta = \left(\frac{S}{32}-2\right)\left(\frac{1}{S}-\frac{2}{q}\right)\left(\operatorname{sinc}\Big(\frac{1}{2}+\frac{1}{2S}\Big)-2\sin(\pi/32)\right)^2, \tag{38}$$

where we used that $\lfloor S/32 - 1 \rfloor \geq (S/32 - 2)$. The measured value ℓ is a multiple of q/S rounded to the nearest integer i.e. $\ell = [mq/S]$ for some m.

Unlike the case of period finding algorithm where the period is integral, the period S of $|\psi\rangle_{o,S}$ cannot be reconstructed with Fourier sampling one (pseudo-periodic) quantum state. However, as shown below, we can reconstruct using the method suggested by Hallgren in [4]. Suppose we have two measurements $c = [kq/S]$ and $d = [lq/S]$, obtained by Fourier sampling the pair of periodic states, then k/l occurs as a convergent of c/d and we can compute an integer close to S by computing $[kq/c]$. Without loss of generality assume that $0 < k \leq l < \lfloor S/32 \rfloor$. Assume that $c = kq/S + \epsilon_c$ and $d = lq/S + \epsilon_d$ where $-1/2 \leq \epsilon_c, \epsilon_d \leq 1/2$. Then

$$\begin{aligned}\left|\frac{c}{d}-\frac{k}{l}\right| &= \left|\frac{kq+\epsilon_c S}{lq+\epsilon_d S}-\frac{k}{l}\right| = \left|\frac{S(\epsilon_c l-\epsilon_d k)}{l^2 q+\epsilon_d S l}\right|\\ &\leq \left|\frac{S(l+k)/2}{l^2q-Sl/2}\right| \leq \left|\frac{Sl}{l^2q-Sl/2}\right|\\ &= \left|\frac{1}{lq/S-1/2}\right| < \frac{1}{2l^2},\end{aligned}$$

under the assumption that $0 < k \leq l < \lfloor S/32 \rfloor$ and $q \geq S^2$. Thus k/l is a convergent of c/d. Since $l \leq \lfloor S/32 \rfloor$, we only need to compute the convergents c_i/d_i whose denominators d_i are less than $\lfloor q/32 \rfloor$. We now form the list of candidate estimates for S as

$$\mathcal{L} = \Big\{ [c_i q/c] \mid d_i \leq \lfloor q/32 \rfloor \Big\}. \tag{39}$$

As the d_i grow exponentially, $|\mathcal{L}| = O(\text{polylog}(|S|))$.

Since k/l is a convergent of c/d, we know that there exists an estimate $\hat{S} = [kq/c] \in \mathcal{L}$. We now show that $\hat{S}$ satisfies $|S - \hat{S}| \leq 1$. Let $c = kq/S + \epsilon_c$ and $\hat{S} = kq/c$, where $|\epsilon_c| \leq 1/2$. Then, we can bound $|S - \hat{S}|$ as

$$\begin{aligned}
|S - \hat{S}| &= \left| S - \frac{S}{1 + \epsilon_c S/kq} \right| \leq \left| \frac{\epsilon_c S^2/kq}{1 + \epsilon_c S/kq} \right| \\
&\leq \left| \frac{\epsilon_c/k}{1 + \epsilon_c/kS} \right| \text{ because } q \geq S^2 \\
&= \left| \frac{\epsilon_c}{k} \right| \left| \frac{1}{1 + \epsilon_c/kS} \right| \\
&\leq \frac{1}{2k} \frac{1}{1 - 1/2kS} \leq \frac{1}{2k} \cdot 2 \\
&\leq 1.
\end{aligned}$$

We now compute a lower bound on the success probability of the algorithm. We have already seen that the probability of a pair of good measurements is given by (38). In order to be able to recover the period S, we require k and l to be coprime. By Lemma A.1, the probability that k, l are coprime is at least $1/2$. Thus the overall success probability of the algorithm is greater than $\beta^2/2 = \Omega(1)$ $\square$.

The algorithm does not return a single value for the period but rather a small list of candidates for the period. This presumes a post processing step by which we can single out the solutions.

Further, we note that the previous algorithm uses a pair of pseudo-periodic states and if these states are being prepared probabilistically, then we must factor that into the success probability of the algorithm.

4 Quantum Algorithm for Approximating the Circumference

Our goal is to set up pseudo-periodic states whose period is a multiple of the circumference of an infrastructure. Then the quantum algorithm of the preceding section can be applied to extract an integer close to the circumference. With this knowledge, the circumference can be computed to the desired accuracy by a classical algorithm.

4.1 Pseudo-periodic states from infrastructures

In section 2.6, we showed that an approximate version $\tilde{h}$ of h can be computed so that properties **P1**, **P2** are satisfied. For this approximate version to be useful, it is necessary

that the f-representations at the evaluation points meet the condition stated in Eq. (14). In this subsection, we show how to satisfy this condition which allows us to compute $\tilde{h}$ so that the first component is always correct and the error in the second component is under control. However, $\tilde{h}$ does not induce the periodic states that we discussed in the previous section. To create a periodic quantum state it is essential to work with a "quantized" version of $\tilde{h}$. Therefore we introduce the function $h_N : \mathbb{Z} \to X \times \mathbb{Z}$ by setting

$$h_N(i) = (\tilde{x}, \lfloor \tilde{f} N \rfloor), \tag{40}$$

where $\tilde{h}(\frac{i}{N} + \frac{j}{L}) = (\tilde{x}, \tilde{f})$. When **P2** is satisfied, it is helpful to interpret h_N in the following way: $h_N(i) = (x, k)$, then k is the number of sampling points between $d(x) + \lfloor (i/N + j/L)/R \rfloor R$ and $i/N + j/L$.

The incorrectness in $\tilde{h}$ cannot be avoided if the evaluation points r are chosen arbitrarily. As already stated in Lemma 14, we assume that the evaluation points are of the form k/L for some large integer L and bounded k. Even so, we cannot always evaluate $\tilde{h}$ correctly for every k. Therefore, we further restrict the evaluation of $\tilde{h}$ to a subset of the points which are $\frac{1}{N}$ uniformly spaced along a bounded interval, where N divides L. We choose $N \geq \lceil 2/d_{\min} \rceil$ so that there are at least two evaluation points $\frac{i}{N}$ and $\frac{i+1}{N}$ between any two adjacent elements of $\mathcal{I}$. This is shown in figure 2. The dashed lines indicate the sampling points.

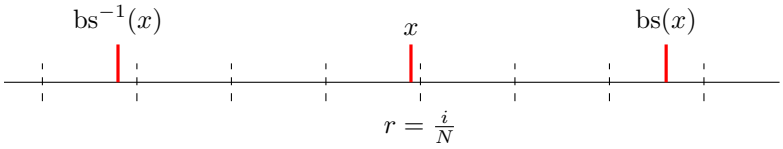

Fig. 2. Evaluating $\tilde{h}$ on a discrete set of uniformly spaced points.

But this is still inadequate to satisfy Eq. (14), as some of the evaluation points could be very close to elements of the infrastructure. So we shift all the evaluation points by a random offset of the form $\frac{j}{L}$, where j is chosen uniformly at random from $\{0, 1, \ldots, \frac{L}{N} - 1\}$. This is shown in figure 3. The solid lines indicate the shifted evaluation points and they are still of the form k/L.

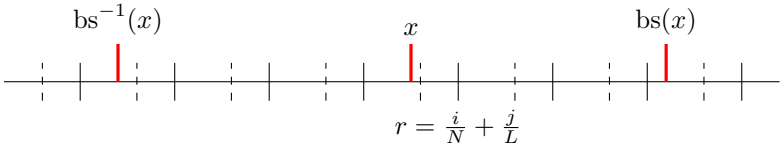

Fig. 3. The evaluation points are shifted by a random offset j/L so that none of them are too close to the elements of the infrastructure.

Now we can show that with high probability Eq. (14) is satisfied and can use Lemma 14 to guarantee that $\tilde{h}$ can be computed with the precision stated in Eq. (15).

Lemma 20: Let $N \geq \lceil 2/d_{\min} \rceil$. Suppose we evaluate h_N at points $i/N + j/L$ for $i \in \{0, \ldots, q-1\}$, where j chosen uniformly at random from $\{0, 1, \ldots, L/N - 1\}$, and L is an

integer such that

$$L \geq N \left\lceil \frac{2\bar{k}}{(1-p_h)} \left\lceil \frac{q}{N d_{\bar{k}}} \right\rceil \right\rceil. \tag{41}$$

Then with probability greater or equal to p_h, no sampling point $i/N + j/L$ is closer than $1/L$ to any element x of the $\mathcal{I}$, i.e.,

$$|(d(x) - i/N - j/L) \bmod R| \geq 1/L. \tag{42}$$

Proof: By assumption **A7** there are at most $\bar{k}\lceil q/Nd_{\bar{k}}\rceil$ elements of $\mathcal{I}$ in the interval $[0, q/N]$. There are L/N possible offsets to choose from. Since the offsets are spaced at $1/L$, any element $x \in \mathcal{I}$ can be within a distance of less than $1/2L$ for at most two offsets. The fraction of offsets that are not useful is given by $2\bar{k}\lceil q/Nd_{\bar{k}}\rceil/(L/N) \leq 1-p_h$ provided that L is chosen as in Eq. (41) $\square$.

When L is chosen according to Lemma 20, we have $h_N(i) = (x, \lfloor \tilde{f}N \rfloor)$, where $\tilde{h}(\frac{i}{N} + \frac{j}{L}) = (x, \tilde{f})$. We use x instead of $\tilde{x}$ on purpose to emphasize again that the first component is correct. It is crucial to observe that $\lfloor \tilde{f}N \rfloor$ is equal to $\lfloor fN \rfloor$. This is because **P2** holds and no evaluation point is within $1/L$ of any element of the infrastructure.

The preceding results imply that $h_N(i)$ can be computed efficiently and correctly.

Corollary 21: If Lemma 20 holds, then for all i with $0 \leq i \leq 2N^2R^2$ the value $h_N(i) = (\tilde{x}, \lfloor \tilde{f}N \rfloor)$ is equal to $(x, \lfloor fN \rfloor)$, where $\tilde{h}(i/N+j/L) = (\tilde{x}, \tilde{f})$ and $h(i/N+j/L) = (x, f)$.

Next we show that h_N when evaluated over a finite interval induces a periodic state with probability greater than or equal to $1/2$, if we assume that no sampling point is too close to any element of the infrastructure.

Lemma 22: Let $N \geq \lceil 2/d_{\min} \rceil$ and let $|\psi\rangle = q^{-1/2} \sum_{i=0}^{q-1} |i\rangle |h_N(i)\rangle$. We assume that no element of the infrastructure is too close to the sampling points $i/N + j/L$, where j and L are chosen as in Lemma 20. Then, with probability greater than

$$p_{\text{periodic}} = \left(1 - \frac{1}{Nd_{\min}} - \frac{1}{NR}\right)\left(1 - \frac{2NR}{q}\right) \tag{43}$$

measuring the second register of $|\psi\rangle$ induces a periodic state with period NR,

$$|\psi\rangle_{k,NR} = \frac{1}{\sqrt{p}} \sum_{\ell=0}^{p-1} |\lfloor k + \ell NR \rceil\rangle, \tag{44}$$

where p is equal to one of the values[e] $\lfloor q/NR \rfloor - 1$, $\lfloor q/NR \rfloor$, or $\lfloor q/NR \rfloor + 1$.

Proof: Denote the measurement outcome by (x, m). First, we show that if (x, m) satisfies a certain condition, then the resulting post-measurement state is a pseudo-periodic state. Second, we estimate the probability that we obtain such measurement outcome.

[e] Note that $N \geq \lceil 2/d_{\min} \rceil$, implies that $NR > 2$, and therefore, p must be at least $\lfloor q/NR \rfloor - 1$.

Assume that $h_N(k) = (x, m)$ for some $k \in \{0, \ldots, \lfloor NR \rfloor\}$. Then, in ℓth period the sampling points are at a distance $\alpha_\ell + m_\ell/N$ for $m_\ell \in \{0, 1, \ldots, \lfloor N \operatorname{bs}(x) \rfloor\}$ from the element x. This is illustrated in figure 4. Under the assumption of Lemma 20, $1/L \leq \alpha_\ell \leq 1/N - 1/L$.

Consider now the sampling points for the zeroth period and some other period $\ell \neq 0$.

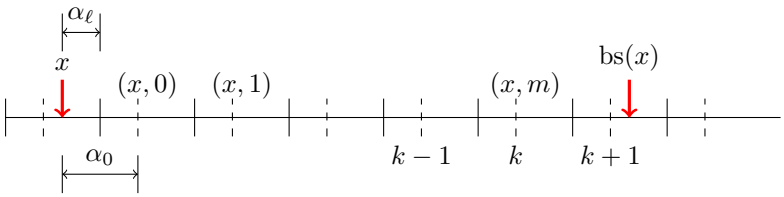

Fig. 4. Evaluation points of $h_N(k)$ in the ℓth period.

Then, the following cases arise: $1/L \leq \alpha_\ell \leq \alpha_0$, and $\alpha_0 < \alpha_\ell \leq 1/N - 1/L$. As can be seen from the figure above, if $1/L \leq \alpha_\ell \leq \alpha_0$, then we must have $h_N(k) = h_N(k + \lfloor \ell NR \rfloor)$. On the other hand, if $\alpha_0 < \alpha_\ell \leq 1/N - 1/L$, then it is clear that $h_N(k) = h_N(k + \lceil \ell NR \rceil)$ unless k corresponds to the last sampling point between the elements x and $y = \operatorname{bs}(x)$ since in this case $h_N(\lceil k + \ell NR \rceil) = (y, 0) \neq h_N(k)$.

On the one hand, if k does not correspond to the last sampling point between two adjacent elements of $\mathcal{I}$, then for all $\ell \in \{0, 1, \ldots, p-1\}$ we have $h_N(k + \lfloor \ell NR \rceil) = h_N(k)$. On the other hand, if k corresponds to the last evaluation point between two elements, then the preimage may not contain all ℓ.

We now estimate the probability of obtaining an outcome (x, m) such that $h_N(k) = (x, m)$ and the offset $k \in \{0, \ldots, \lfloor NR \rfloor\}$ does not correspond to the last evaluation point between any two elements.

There are $\lfloor NR \rfloor + 1$ possible offsets in the zeroth period. At most $\lceil R/d_{\min} \rceil$ of these can correspond to last evaluation points between two elements. We know that the preimage of a "good" measurement outcome (x, m) contains at least $\lfloor q/NR \rfloor - 1$ elements. So, the probability of obtaining a good measurement outcome is at least

$$\begin{aligned} p_{\text{periodic}} &= \frac{(\lfloor NR \rfloor + 1 - \lceil R/d_{\min} \rceil) \cdot (\lfloor q/NR \rfloor - 1)}{q} \\ &\geq (NR - R/d_{\min} - 1)(q/NR - 2)/q \\ &= \left(1 - \frac{1}{Nd_{\min}} - \frac{1}{NR}\right)\left(1 - \frac{2NR}{q}\right) \end{aligned}$$

□.

4.2 Presentation and proof of the quantum algorithm

Theorem 23: (Estimating the circumference to arbitrary accuracy) Let $\mathcal{I}$ be an infrastructure satisfying the assumptions **A**1–**A**7. For any $\delta > 0$, there is an efficient Las Vegas algorithm that outputs an estimate $\hat{R}$ of the circumference R of $\mathcal{I}$ such that $|R - \hat{R}| \leq \delta$.

Let $N \geq \lceil 2/d_{\min} \rceil$, $S = NR$, p_h the probability of evaluating h_N correctly, and p_{periodic} the probability of creating a periodic state, see Eq. (43). Then, the classical algorithm invokes Algorithm 1 an expected $O(1/q_{\text{success}})$ number of times, where q_{success} is

$$q_{\text{success}} \geq \frac{p_h^2 p_{\text{periodic}}^2}{2} \left(\frac{1}{32} - \frac{2}{S}\right)^2 \left(1 - \frac{2S}{q}\right)^2 \left(\operatorname{sinc}\left(\frac{1}{2} + \frac{1}{2S}\right) - 2\sin(\pi/32)\right)^4 . \tag{45}$$

The classical computations take $\text{poly}(\log(R), \log(1/\delta))$ time.

Proof: We first create an pseudo-periodic state in $\mathbb{C}^q$, where q is chosen as specified by Algorithm 1. We create the superposition

$$|\psi\rangle = \frac{1}{\sqrt{q}} \sum_{i=0}^{q-1} |i\rangle |h_N(i)\rangle.$$

If the conditions of Lemma 20 are satisfied, then $|\psi\rangle$ will be created correctly with a probability p_h. Then by Lemma 22, measuring the second register of the state results in a periodic state $|\psi\rangle_{k,NR}$ with probability $\geq 1/2$, where $p \in \{\lfloor q/S \rfloor - 1, \lfloor q/S \rfloor, \lfloor q/S \rfloor + 1\}$. Algorithm 1 returns $\mathcal{L}$, a list of candidates for S, which contains an element $\hat{S}$ which satisfies $|S - \hat{S}| \leq 1$. The probability of this event is

$$\Pr(|S - \hat{S}| \leq 1) \geq p_h^2 p_{\text{periodic}}^2 p_{\text{success}} , \tag{46}$$

where p_{success} is defined in Eq. (32). The factor of $p_h^2 p_{\text{periodic}}^2$ is due to the fact that the Algorithm 1 needs to create a pair of the pseudo-periodic states.

Assume that $|S - \hat{S}| \leq 1$ is present (of course, we do not know this). This is equivalent to $|R - R'| \leq 1/N$, where $R' = \hat{S}/N$. We actually check for a slightly weaker condition namely, $|(R - R') \bmod R| \leq 1/N$. But this suffices.

Recall that we always choose $N \geq \lceil 2/d_{\min} \rceil$. This implies that either $h(R') = (x_0, f)$ with $f \leq \frac{1}{N}$ or $h(R') = (\text{bs}^{-1}(x_0), g)$ with $g \geq \Delta_{\text{bs}}(\text{bs}^{-1}(x_0)) - 1/N$. If we evaluate $\tilde{h}$, the approximate version of h, at R' with precision $\delta_{\text{prec}} \leq \frac{1}{2N}$, then it remains the case that we can only obtain either $(x_0, \tilde{f})$ or $(\text{bs}^{-1}(x_0), \tilde{g})$. If so we can conclude that $|R - R' \bmod R| \leq 1/N$.

Now assume that $|(R - R') \bmod R| > 1/N$ holds. In this case, we may or may not encounter $\text{bs}^{-1}(x_0)$ or x_0 by evaluating $\tilde{h}$ at R'.

Because our test actually checked for $|(R - R') \bmod R| \leq 1/N$, we could have some spurious solutions when R' is a multiple of R. If this is the case, then we return the smallest such R' as satisfying $|R - R'| \leq 1/N$. We then obtain an estimate for R as follows.

Once we have encountered $\text{bs}^{-1}(x_0)$ or x_0, we can compute $\tilde{h}(R')$ with precision $\delta/2$. If we obtain $(\text{bs}^{-1}(x_0), \tilde{g})$, then we set

$$\hat{R} = R' - \tilde{g} + \Delta_{\text{bs}}(\text{bs}^{-1}(x_0)) , \tag{47}$$

where we compute the distance Δ_{bs} with precision $\delta/2$. If we obtain $(x_0, \tilde{f})$, then we set $\hat{R} = R' - \tilde{f}$. All these computations can be carried out in $\text{poly}(\log(R), \log(1/\delta)$ time.

The expected number of times we have to invoke the quantum algorithm to encounter $\mathrm{bs}^{-1}(x_0)$ or x_0 is clearly at most $1/q_{\text{success}}$ $\square$.

There is a subtle point worth spelling out. In each run of the algorithm, there are two evaluations of h_N. We assume that the same random shift is used in both these evaluations and in any subsequent $O(1/q_{\text{success}})$ runs. Only if the algorithm fails in all these runs do we change the offset and repeat the process.

Finally, it can be easily verified for sufficiently large S, say $S \geq 256$, the lower bound on the success probability is greater than a constant, irrespective of the size of the problem.

The proposed algorithm when specialized to number fields improves upon [4] in the following aspects. The probability of success of the our algorithm is bounded from below by Eq. (32) which is a constant 10^{-5}. This is in contrast to [4] where the guaranteed success probability decreases as $\Omega(1/\log^4(M))$ ($M \geq NR$) and is always less or equal to 10^{-9} (see [4, Claim 3.5 and Lemma 3.4]). Our guaranteed success probability is also better than the one in [6], which is only 2^{-26}. Our higher guaranteed success probability implies that fewer repetitions are required to boost the success probability to any desired level, thereby, leading to lower gate complexity of the algorithm. In addition, our proposed algorithm requires a smaller Quantum Fourier transform, thereby also lowering the space and time complexities.

5 Quantum Algorithm for Solving the Generalized Discrete Logarithm Problem in Infrastructures

In this section we give a quantum algorithm for the discrete logarithm problem. Given an element x of an infrastructure $\mathcal{I} = (X, d)$ we are required to find the distance of x, namely $d(x)$.

The function that is of interest in the computation of the discrete log problem is given by $g(a, b) : \mathbb{Z} \times \mathbb{R} \to \mathcal{I} \times \mathbb{R}$ where $g(a, r) = a \cdot (x, 0) + h(r)$. By Lemma 15 we can compute the approximate version $\tilde{g}$ of g, so that it satisfies properties **P1**, and **P2**.

As in the circumference case, we evaluate $\tilde{g}$ at carefully selected points to ensure that the first component is always correct and quantize the second component. This resulting function is $g_N(a, b) : \mathbb{Z} \times \mathbb{Z} \to \mathcal{I} \times \mathbb{Z}$

$$g_N(a, b) = \left(\tilde{y}, \left\lfloor \tilde{f} N \right\rfloor\right), \tag{48}$$

where $\tilde{g}(a, b/N + j/L) = (\tilde{y}, \tilde{f})$.

The first component of g_N is correct provided that Eq. (14) is satisfied for all evaluation points of g_N, i.e., none of the evaluation points are closer than $1/L$ to any element of the infrastructure. As in the case of h_N, we achieve this with high probability by applying a random shift of the form j/L. The following lemma shows how to find a suitable L.

Lemma 24: (Offset for DLOG) Suppose $\mathcal{I}$ is an infrastructure that satisfies the assump-

tions **A**1-7. Let $\mathcal{A} \subseteq \{0, 1, \ldots, A-1\}$ and $\mathcal{B} \subseteq \{0, 1, \ldots, \lfloor RN \rfloor - 1\}$. Let

$$L \geq \left\lceil \frac{2A\bar{k}}{(1-p_g)} \left\lceil \frac{1}{d_{\bar{k}}} \left(R - \frac{1}{N} \right) \right\rceil \right\rceil N. \tag{49}$$

Let $j \in \{0, 1, \ldots, L/N - 1\}$ be chosen uniformly at random. Then, the probability that

$$\left| (ad_x + \frac{b}{N} + \frac{j}{L} - d_y) \bmod R \right| \geq \frac{1}{L} \tag{50}$$

holds for all $(a, b) \in \mathcal{A} \times \mathcal{B}$ and all $y \in X$ is greater or equal to p_g.

Proof: Consider a fixed $a \in \mathcal{A}$, then all the points $ad_x + b/N + j/L$ are contained in the interval $[ad_x + j/L, ad_x + (\lfloor RN - 1 \rfloor)/N + j/L]$. This interval contains at most $\bar{k}\lceil (R - 1/N)/d_{\bar{k}} \rceil$ elements $y \in X$ since its length is $\lfloor RN - 1 \rfloor / N \leq (R - 1/N)$. Observe that no $y \in X$ can be closer than $1/L$ to any evaluation point of the above form for more than two offsets.

Hence, if we consider all $a \in \mathcal{A}$, then at most $2A\bar{k}\lceil (R - 1/N)/d_{\bar{k}} \rceil$ offsets are bad. Assuming L as stated above, this implies that the probability that there is at least one element and at least one evaluation point that are closer than $1/L$ to each other is at most $(2A\bar{k}(R - 1/N)/d_{\bar{k}})/(L/N) \leq 1 - p_g$ □.

We always compute $\hat{R}$ with sufficiently high precision so that $|\hat{R} - R| < 1/(2N)$ holds. Then, we have $\hat{R} > R - 1/2N$ and a suitable choice for L would be $\left\lceil 2A\bar{k} \left\lceil \hat{R}/d_{\bar{k}} \right\rceil / (1 - p_g) \right\rceil N$.

In the quantum algorithm for approximating the circumference, we encounter superpositions of the form:

$$|\psi\rangle = \frac{1}{\sqrt{|\mathcal{A}_{x,m}|}} \sum_{a \in \mathcal{A}_{x,m}} |a\rangle |(x, m)\rangle,$$

where $\mathcal{A}_{x,m}$ has the special form $\{\lfloor k + jRN \rceil \,:\, j = 0, \ldots, p\}$ and (x, m) is equal to $h_N(k)$.

A somewhat similar type of quantum state appears in the discrete logarithm problem. A major difference is that it involves a function of two variables

$$|\psi\rangle = \frac{1}{\sqrt{|\mathcal{A}_{y,\ell}|}} \sum_{(a,b) \in \mathcal{J}} |a\rangle |b\rangle |(y, \ell)\rangle,$$

where $\mathcal{A}_{y,\ell}$ is now the preimage of $(y, \ell) \in \operatorname{im} g_N$, i.e., $g_N(a, b) = (y, \ell)$ for $(a, b) \in \mathcal{A}_{y,\ell}$.

The intuition based on Lemma 12, which characterizes the kernel of the perfect function g, suggests that the elements in $\mathcal{A}_{y,\ell}$ lie "close" to a line whose slope encodes the distance of the element x. This statement is proved in Lemma 25, which establishes the exact relation between a and b for g_N. Lemma 26 establishes upper and lower bounds on the size of the preimage of (y, ℓ).

The intuition based on the quantum algorithm for the discrete logarithm problem in finite cyclic groups suggests that we can extract the slope by Fourier sampling. This statement is

proved in Theorem 27.

Lemma 25: Let $\emptyset \neq \mathcal{A} \subseteq \{0, 1, 2, \ldots, A-1\}$ where A is a positive integer and $\mathcal{B} \subseteq \{0, 1, \ldots, \lfloor RN \rfloor - 1\}$. Denote by $g_N(\mathcal{A} \times \mathcal{B})$ the image of the function g_N, i.e.,

$$g_N(\mathcal{A} \times \mathcal{B}) = \{g_N(a, b) \,:\, a \in \mathcal{A}\,,\, b \in \mathcal{B}\}\,. \tag{51}$$

For each $(y, \ell) \in g_N(\mathcal{A} \times \mathcal{B})$, the preimage $g_N^{-1}(y, \ell)$ has the form

$$g_N^{-1}(y, \ell) = \{(a, b_a) \,:\, a \in \mathcal{A}_{y,\ell}\}\,, \tag{52}$$

where $\mathcal{A}_{y,\ell} \subseteq \mathcal{A}$ and assuming that a random shift of j/L has been applied to the evaluation points, the values b_a satisfy the condition

$$\left\lfloor \frac{ad_x + \frac{b_a}{N} + \frac{j}{L}}{R} \right\rfloor R + d_y + \gamma_a + \frac{\ell}{N} = ad_x + \frac{b_a}{N} + \frac{j}{L} \tag{53}$$

with $1/L \leq \gamma_a \leq 1/N - 1/L$. The cardinality of the image satisfies the inequalities

$$|\mathcal{B}| \leq |g_N(\mathcal{A} \times \mathcal{B})| \leq \lfloor R(N + 1/d_{\min}) \rfloor\,. \tag{54}$$

Proof: Let $(y, \ell) \in g_N(\mathcal{A} \times \mathcal{B})$ be arbitrary. Suppose that $(a, b_a) \in g_N^{-1}(y, \ell)$. Then we must have

$$\begin{aligned} d_y + \frac{\ell}{N} + \gamma_a &\equiv ad_x + \frac{b_a}{N} + \frac{j}{L} \bmod R \\ &= ad_x + \frac{b_a}{N} + \frac{j}{L} - \left\lfloor \frac{ad_x + \frac{b_a}{N} + \frac{j}{L}}{R} \right\rfloor R, \end{aligned}$$

where $1/L \leq \gamma_a \leq 1/N - 1/L$. This constraint on γ_a is due to the fact that none of the sampling points are within a distance of less than $1/L$ from the elements of the infrastructure.

The second component ℓ is bounded from above by

$$\ell \leq N\Delta_{\mathrm{bs}}(y)$$

since the inequality

$$\left\lfloor \frac{ad_x + \frac{b_a}{N} + \frac{j}{L}}{R} \right\rfloor R + d_y + \gamma_a + \frac{\ell}{N} < \left\lfloor \frac{ad_x + \frac{b_a}{N} + \frac{j}{L}}{R} \right\rfloor R + d_y + \Delta_{\mathrm{bs}}(y)$$

holds for all (a, b_a) with $g_N(a, b_a) = (y, \ell)$. This implies that the number of images whose first component is equal to y is at most $N\Delta_{\mathrm{bs}}(y) + 1$. Summing over all elements of the infrastructure yields the upper bound $RN + R/d_{\min}$. We can improve this to $\lfloor R(N + 1/d_{\min}) \rfloor$ since the cardinality of $g_N(\mathcal{A} \times \mathcal{B})$ is an integer. Hence, $|g_N(\mathcal{A} \times \mathcal{B})| \leq \lfloor R(N + 1/d_{\min}) \rfloor$ $\square$.

A condition similar to Eq. (54) has been established in [4] for the principal ideal problem. The condition as derived in [4] may not be satisfied for some infrastructures. Therefore, we relax this constraint and clarify certain crucial assumptions on the size of the preimage in Lemma 26.

Lemma 26: Let $\mathcal{A}$ and $\mathcal{B}$ be as in Lemma 25. Consider the probability distribution $p = (p_{y,\ell})$ on $g_N(\mathcal{A} \times \mathcal{B})$ where the probabilities of the elementary events (y, ℓ) are given by

$$p_{y,\ell} = \frac{|g_N^{-1}(y,\ell)|}{|\mathcal{A}||\mathcal{B}|} \,. \tag{55}$$

Let X be the random variable that takes on the value $|g_N^{-1}(y,\ell)|$ if the event (y,ℓ) occurs. Then, we have

$$\Pr\left(X \geq \kappa|\mathcal{A}|\right) \geq \frac{1}{1-\kappa}\left(\frac{|\mathcal{B}|}{\lfloor R(N+1/d_{\min})\rfloor} - \kappa\right) \tag{56}$$

for any $\kappa \in (0,1)$. The expected value $\mathbb{E}[X]$ is bounded from below by

$$\begin{aligned}
\mathbb{E}[X] &= \sum_{(y,\ell)} p_{y,\ell}\, |g_N^{-1}(y,\ell)| \\
&= |\mathcal{A}||\mathcal{B}| \sum_{(y,\ell)} p_{y,\ell}^2 \\
&\geq |\mathcal{A}||\mathcal{B}|\, \frac{1}{|g_N(\mathcal{A}\times\mathcal{B})|} \\
&\geq |\mathcal{A}||\mathcal{B}|\, \frac{1}{\lfloor R(N+1/d_{\min})\rfloor} \,.
\end{aligned}$$

We used that the sum $\sum p_{y,\ell}^2$ is minimized when the probability distribution is uniform over $g_N(\mathcal{A}\times\mathcal{B})$ and that $|g_N(\mathcal{A}\times\mathcal{B})| \leq \lfloor R(N+1/d_{\min})\rfloor$.

Let $t = \Pr(X \geq \kappa\, \mathbb{E}[X])$. Then, we must have

$$t|\mathcal{A}| + (1-t)\kappa|\mathcal{A}| \geq \mathbb{E}[X] \geq |\mathcal{A}|\,|\mathcal{B}|/\lfloor R(N+1/d_{\min})\rfloor$$

since X is bounded by $|\mathcal{A}|$ from above. The desired lower bound on t follows now easily

Theorem 27: Let $\mathcal{I}$ be an infrastructure containing at least 3 elements and satisfying the axioms **A1**–**A7**. For all $x \in X$, Algorithm 2 returns an integer $\hat{d}_x$ such that $|d_x - \hat{d}_x| \leq 1$, where d_x is the distance of x.

Let p_g be the probability of correctly evaluating g_N and κ a real number with $(1-\mathrm{sinc}(3/4))/(1-2\sin(\pi/32)) < \kappa < 1 - 2/(2\mathrm{q}+1)$. Then, the success probability of the algorithm is $\Omega(1)$ and at least

$$p_g \max_{\kappa}\left(1 - \frac{2}{(2q+1)(1-\kappa)}\right)^2 \frac{\kappa^2}{2}\left(1 - 2\sin(\frac{\pi}{32}) - \frac{(1-\mathrm{sinc}(3/4))}{\kappa}\right)^4 \left(\frac{1}{64} - \frac{2}{B}\right)^2 \tag{57}$$

Algorithm 2 GENERALIZED DISCRETE LOGARITHM.

1: Choose $M \geq \lceil 2R+1 \rceil$.
2: Determine $\hat{R}$ and N such that $\left| M \left\lfloor \hat{R}N \right\rceil - MRN \right| \leq 1/2$ and $N = q\lceil 2/d_{\min} \rceil$ for a positive integer with $q \leq 4M$. Set $B = \left\lfloor \hat{R}N \right\rceil$ and $A = MB$.
3: Choose $L = \left\lceil 2A\bar{k} \left\lceil \hat{R}/d_{\bar{k}} \right\rceil / (1-p_g) \right\rceil N$.
4: Evaluate g_N in superposition over $\{0, 1 \ldots, A-1\} \times \{0, 1, \ldots, B-2\}$ twice.
5: Fourier sample over $\mathbb{Z}_A \times \mathbb{Z}_B$ to obtain (h_1, k_1) and (h_2, k_2).
6: Find integers s, t such that $sk_1 + tk_2 = 1$, using the extended Euclidean algorithm.
7: Compute $r = \frac{sh_1 + th_2}{NM}$
8: Return $\hat{d}_x = r - \left\lfloor r/\hat{R} \right\rceil \hat{R}$.

where $B = \left\lfloor \hat{R}N \right\rceil$ and q is chosen as in Algorithm 2.

Proof: We compute an estimate $\hat{R}$ such that

$$|R - \hat{R}| \leq \epsilon \leq \frac{1}{16M^2\lceil 2/d_{\min} \rceil}\,. \tag{58}$$

We now show that there is an efficient method that determines positive integers $B = \left\lfloor \hat{R}N \right\rceil$ and N such that

$$|MB - MNR| \leq \frac{1}{2}\,. \tag{59}$$

To do this, we bound this deviation by

$$\begin{aligned} |MB - MNR| &= |M\lfloor N\hat{R} \rceil - MN\hat{R} + MN\hat{R} - MNR| && (60)\\ &\leq |M\lfloor N\hat{R} \rceil - MN\hat{R}| + MN\epsilon && (61) \end{aligned}$$

The efficient method in Lemma A.2 gives us a convergent p/q with $q \leq 4M$ such that

$$\left| \frac{p}{q} - \hat{R}\lceil 2/d_{\min} \rceil \right| \leq \frac{1}{4Mq}\,. \tag{62}$$

The numerator p has the form $\left\lfloor \hat{R}\lceil 2/d_{\min} \rceil q \right\rceil$. The bound in Eq. (62) and the form of the numerator directly imply that $N = q\lceil 2/d_{\min} \rceil$ has the desired properties. Both terms in Eq. (61) are smaller than $1/4$ for this choice.

Observe that $B - 2 = \left\lfloor \hat{R}N \right\rceil - 2 \leq \lfloor RN \rfloor - 1$ because $\hat{R}$ has been computed with such high precision. We define the sets $\mathcal{B} = \{0, 1, \ldots, B-2\}$ and $\mathcal{A} = \{0, 1, \ldots, A-1\}$.

We create the superposition

$$\frac{1}{\sqrt{|\mathcal{A}||\mathcal{B}|}} \sum_{a \in \mathcal{A}} \sum_{b \in \mathcal{B}} |a\rangle |b\rangle |g_N(a,b)\rangle\,.$$

We know that with probability greater or equal to p_g all the values $g_N(a,b)$ are correct.

We measure the third register. Denote the outcome by (y,ℓ). Lemma 26, guarantees that $|\mathcal{A}_{y,\ell}| \geq \kappa|\mathcal{A}|$ holds with probability greater or equal to

$$p_\kappa \geq \frac{1}{1-\kappa}\left(\frac{|\mathcal{B}|}{\lfloor R(N+1/d_{\min})\rfloor} - \kappa\right). \tag{63}$$

Since $N = q\lceil 2/d_{\min}\rceil$, we can bound p_κ

$$\begin{aligned} p_\kappa &\geq \frac{1}{1-\kappa}\left(\frac{NR-3}{NR(1+1/2q)} - \kappa\right), &(64)\\ &\geq \frac{1}{1-\kappa}\left(\frac{2q-1}{2q+1} - \kappa\right) = 1 - \frac{2}{(2q+1)(1-\kappa)}, &(65)\end{aligned}$$

where we used the assumption that $\mathcal{I}$ has at least 3 elements and therefore $R > 3d_{\min}$, and $NR > 6q$.

Lemma 25 implies that the post-measurement state has the form

$$\frac{1}{\sqrt{|\mathcal{A}_{y,\ell}|}} \sum_{a\in\mathcal{A}_{y,\ell}} |a\rangle|b_a\rangle \tag{66}$$

and there exists a unique b_a for each $a \in \mathcal{A}_{y,\ell}$ such that

$$b_a = -ad_xN + \left\lfloor \frac{ad_x + \frac{b_a}{N} + \frac{j}{L}}{R} \right\rfloor RN + d_yN + \gamma_aN + \ell - \frac{jN}{L}, \tag{67}$$

where $1/L \leq \gamma_a \leq 1/N - 1/L$. We rewrite the condition on b_a as

$$b_a = -ad_xN + \left\lfloor \frac{ad_x + \frac{b_a}{N} + \frac{j}{L}}{R} \right\rfloor RN + \gamma_aN + \Delta, \tag{68}$$

where $\Delta = d_yN + \ell - jN/L$ is constant.

We apply the quantum Fourier transform over $\mathbb{Z}_A \times \mathbb{Z}_B$ to the first registers and obtain the superposition

$$\frac{1}{\sqrt{A}}\frac{1}{\sqrt{B}} \sum_{h\in\mathcal{A}} \sum_{k=0}^{B-1} \frac{1}{\sqrt{|\mathcal{A}_{y,\ell}|}} \sum_{a\in\mathcal{A}_{y,\ell}} \omega_A^{ah+Mbk} |h\rangle|k\rangle. \tag{69}$$

The amplitude of the term $|h\rangle|k\rangle$ is given by

$$\frac{1}{\sqrt{A}}\frac{1}{\sqrt{B}}\frac{1}{\sqrt{|\mathcal{A}_{y,\ell}|}} \sum_{a\in\mathcal{A}_{y,\ell}} \omega_A^{ah+Mb_ak}. \tag{70}$$

The exponent of ω_A in the previous equation is

$$ah + Mk\left(-ad_xN + \left\lfloor \frac{ad_x + \gamma_a + \frac{b_a}{N} + j/L}{R} \right\rfloor NR + \gamma_aN + \Delta\right). \tag{71}$$

The term $Mk\Delta$ is independent of a and can be dropped from the exponent since it does not change the probability distribution.

We now show that we obtain a sample (h,k) such that

$$h = kd_xMN - \left\lfloor \frac{kd_x}{R} \right\rfloor MNR + \epsilon_h \text{ with } |\epsilon_h| \leq \tfrac{1}{2} \tag{72}$$

holds with high probability.[f]

As shown previously, N is chosen such that $MNR - M\left\lfloor N\hat{R} \right\rceil = \eta$ with $|\eta| \leq \frac{1}{2}$. To simplify the notation, we use x to denote the distance d_x of the element x throughout the rest of the proof. The exponent of ω_A modulo A is

$$\begin{aligned}
& a\left(kxMN - \left\lfloor \frac{kx}{R} \right\rfloor MNR + \epsilon_h\right) \\
& +Mk\left(-axN + \left\lfloor \frac{ax + \gamma_a + \frac{b_a}{N} + j/L}{R} \right\rfloor NR + \gamma_a N\right) \\
= \quad & \left(k\left\lceil \frac{ax + \gamma_a + \frac{b_a}{N} + j/L}{R} \right\rceil - a\left\lfloor \frac{kx}{R} \right\rfloor\right) MNR + \epsilon_h a + MN\gamma_a k \\
\equiv \quad & \eta\left(k\left\lceil \frac{ax + \gamma_a + \frac{b_a}{N} + j/L}{R} \right\rceil - a\left\lfloor \frac{kx}{R} \right\rfloor\right) + \epsilon_h a + MN\gamma_a k \\
= \quad & \eta\left(k\left(\frac{ax}{R} + \delta_a\right) - a\left(\frac{kx}{R} + \zeta\right)\right) + \epsilon_h a + Mk\gamma_a \\
= \quad & \eta\delta_a k - \eta\zeta a + \epsilon_h a + MN\gamma_a k \\
= \quad & \delta a + \theta_a\,.
\end{aligned} \tag{73}$$

The (constant) factor $\delta := \epsilon_h - \eta\zeta$ in front of a is less than $3/4$ in absolute value ($\epsilon_h \leq \frac{1}{2}$, $\zeta < 1$ and $\eta < \frac{1}{2}$). Assume we measure $k \leq \lfloor B/64 \rfloor - 1$. Then, for each a the term $\theta_a := (\eta\delta_a + MN\gamma_a)k$ is less than $A/32$ in absolute value (since $|\delta_a| < 2$ and $|\gamma_a N| < 1$).

We can now apply Lemma 18, to bound the probability of measuring (h,k) as in Eq. (72); we denote this probability by p_{hk}. Note that A corresponds to n, the summation index a to j and $\mathcal{A}_{y,\ell}$ to the set $\mathcal{J}$ in the Lemma 18.

The probability p_{hk} is bounded from below by

$$p_{hk} \quad \geq \quad \frac{|\mathcal{A}_{y,\ell}|}{AB}\left(1 - 2\sin(\pi/32) - \Big(1 - \operatorname{sinc}(3/4)\Big)\frac{1}{\kappa}\right)^2, \tag{74}$$

[f] The reason that we consider samples that have this particular form is as follows. Rearranging the terms in the exponent we see that the sum is dominated by the terms $ah - (kd_x/R)MNR + k\lfloor(ad_x + \gamma_a + b_a/Nj/L)/R\rfloor MNR$. The exponent can be approximated as $ah - (kd_x/R - \lfloor ad_x/R\rfloor)MNR$. Therefore, the probability of (h,k) which is determined by the geometric sum

$$\frac{1}{AB\mathcal{A}_{y,\ell}}\left|\sum_{a\in\mathcal{A}_{y,\ell}} \omega^{ah+Mb_ak}\right|^2 \approx \frac{1}{AB\mathcal{A}_{y,\ell}}\left|\sum_{a\in\mathcal{A}_{y,\ell}} \omega^{a(h-(\frac{kd_x}{R}-\lfloor\frac{kd_x}{R}\rfloor)MNR}\right|^2$$

is large when $h = (kd_x/R - \lfloor kd_x/R\rfloor)MNR + \epsilon_h$, where ϵ_h is to ensure that h is an integer.

where c_δ is as in Lemma 18.

The probability of any good pair (h,k) (with the restriction $k \leq \lfloor B/64 \rfloor - 1$) is bounded from below by

$$\kappa \left(1 - 2\sin(\pi/32) - \Big(1 - \operatorname{sinc}(3/4)\Big)\frac{1}{\kappa}\right)^2 \left(\frac{1}{64} - \frac{2}{B}\right), \tag{75}$$

where we used that $|\mathcal{A}_{y,\ell}| \geq \kappa |A|$ and $\lfloor B/64 - 1 \rfloor \geq B/64 - 2$.

We now show how to obtain an estimate of the distance of x from two good pairs (h_1,k_1) and (h_2,k_2) with the additional restriction that k_1, k_2 are coprime. This is based on the method in [4]. We have $h_i = k_i x NM - \lfloor k_i x/R \rfloor RNM + \epsilon_i$ with $|\epsilon_i| \leq \frac{1}{2}$. Since k_1, k_2 are coprime we know there exist integers s,t such that $sk_1 + tk_2 = 1$, which can be computed by the extended Euclidean algorithm. Let $r = (sh_1 + th_2)/MN$, then we have

$$\begin{aligned}
\frac{sh_1 + th_2}{MN} &= sk_1 x - s\left\lfloor \frac{k_1 x}{R} \right\rfloor R + \frac{s\epsilon_1}{MN} + tk_2 x - t\left\lfloor \frac{k_2 x}{R} \right\rfloor R + \frac{t\epsilon_2}{MN} \\
&= (sk_1 + tk_2)x - s\left\lfloor \frac{k_1 x}{R} \right\rfloor R - t\left\lfloor \frac{k_2 x}{R} \right\rfloor R + \frac{s\epsilon_1 + t\epsilon_2}{MN} \\
&= x - s\left\lfloor \frac{k_1 x}{R} \right\rfloor R - t\left\lfloor \frac{k_2 x}{R} \right\rfloor R + \frac{s\epsilon_1 + t\epsilon_2}{MN} \\
&= x - mR + \epsilon_r,
\end{aligned}$$

where $\epsilon_r = (s\epsilon_1 + t\epsilon_2)/MN$. Since $|s|, |t| \leq \max\{k_1, k_2\}$, and $k_1, k_2 \leq \left\lceil \hat{R}N \right\rceil /32$, it follows that $\epsilon_r = \frac{s\epsilon_1 + t\epsilon_2}{MN} \leq \frac{\lfloor RN \rfloor}{MN} < 1/2$ by our choice of M. Furthermore, $|m| \leq NR/8$, as $|r| \leq 2M\lceil N\hat{R} \rceil \lceil N\hat{R} \rceil / 32MN < NR^2/8$.

We can estimate x by reducing r modulo $\hat{R}$ to bring it within the range $[0, \hat{R})$. This gives us an estimate $\hat{x} = x - m(R - \hat{R}) + \epsilon_r$ and the error $|x - \hat{x}|$ can be bounded as follows:

$$\begin{aligned}
|x - \hat{x}| &\leq |m(R - \hat{R})| + \epsilon_r \leq |m\epsilon| + |\epsilon_r| \\
&\leq \frac{NR}{8} \frac{1}{16M^2 \lceil 2/d_{\min} \rceil} + |\epsilon_r| \leq 1,
\end{aligned}$$

where we used the fact that $M > 2R$ and $N \leq 4M\lceil 2/d_{\min} \rceil$ and $|\epsilon_r| < 1/2$.

The probability of measuring two good samples (h_1,k_1) and (h_2,k_2) such that k_1, k_2 are coprime is given by

$$p_{\text{success}} \geq \frac{1}{2}(p_\kappa p_{hk}(1/64 - 2/B))^2 p_g, \tag{76}$$

where p_g is the probability of evaluating g_N successfully $\square$.

We make the following observations regarding the success probability of the quantum algorithm. First, a simpler lower bound on the success probability can be obtained without having to maximize over κ in Eq. (57), by evaluating this expression at $\kappa = (\kappa_1 + \kappa_2)/2$, where $\kappa_1 = (1 - c_\delta)/(1 - 2\sin(\pi/32))$ and $\kappa_2 = 1 - 2/(2q+1)$. We also note the expression can be further simplified to be completely independent of the size of the infrastructure as follows.

Second, under the assumption that $R \geq 256$ and $q \geq 8$, we can bound $(1/64 - 2/B) \geq 1/128$ and $2/(2q+1) \leq 1/8$, and the lower bound on success probability simplifies to a constant independent of the problem size.

$$\max_{\kappa} p_g \left(1 - \frac{1}{8(1-\kappa)}\right)^2 \frac{\kappa^2}{2} \left(1 - 2\sin(\frac{\pi}{32}) - \frac{(1 - \mathrm{sinc}(3/4))}{\kappa}\right)^4 \left(\frac{1}{128}\right)^2 \tag{77}$$

Although the expressions for the success probability may appear to be a little unwieldy, we hope they provide insight into the various factors affecting the success probability.

Third, we can boost the success probability (strictly speaking, the lower bound on it) by increasing q.

Fourth, we can truly improve upon the success probability by extending the set of usable observations (h_1, k_1) and (h_2, k_2). Currently, we require that $k_i < \lfloor B/64 \rfloor$, but this can be relaxed significantly.

Acknowledgements

We would like to thank Felix Fontein for helpful discussions on infrastructures and suggestions to improve the paper. P.W. thanks Joseph Brennen, Chen-Fu Chiang, and (Raymond) Yiu Yu Ho for helpful discussions. P.S. thanks Robert Raussendorf for his generous support.

P.W. gratefully acknowledges the support from the NSF grant CCF-0726771 and the NSF CAREER Award CCF-0746600. P.S. was sponsored by grants from CIFAR, MITACS and NSERC.

References

1. P. W. Shor, (1997), *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, Siam Journal on Computing, 26:1484–1509.
2. R. Jozsa, (2001), *Quantum factoring, discrete logarithms, and the hidden subgroup problem*, Computing in Science & Engineering, 3:34–43.
3. S. Hallgren, (2002), *Polynomial time quantum algorithm for Pell's equation and the principal ideal problem*, In Proceedings of the 34th Annual ACM Symposium on Theory of Computing, pages 653–658.
4. S. Hallgren, (2007), *Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem*, Journal of the ACM, 54(1):1–19.
5. R. Jozsa, (2003), *Quantum computation in algebraic number theory: Hallgren's efficient algorithm for solving Pell's equation*, Annals. of Physics, 306:241–279.
6. A. Schmidt, (2009), *Quantum algorithms for many-to-one functions to solve the regulator and the principal ideal problem*, arXiv:0912.4807.
7. Arthur Schmidt and Ulrich Vollmer, (2004), *Polynomial time quantum algorithm for the computation of the unit group of a number field*, Technical Report TI-1/04, Technische Universität Darmstadt.
8. A. Schmidt and U. Vollmer, (2005), *Polynomial time quantum algorithm for the computation of the unit group of a number field*, In Proceedings of the 37th Annual ACM Symposium on Theory of Computing, pages 475–480.

9. S. Hallgren, (2005), *Polynomial time quantum algorithm for the computation of the unit group of a number field*, In Proceedings of the 37th Annual ACM Symposium on Theory of Computing, pages 475–480.
10. F. Fontein, (2009), *The infrastructure of a global field and baby step-giant step algorithms*, Dissertation, Universität Zürich.
11. D. Lorenzini, (1996), *An Invitation to Arithmetic Geometry*, volume 9 of Graduate Studies in Mathematics. American Mathematical Society.
12. J. Buchmann and H. C. Williams, (1988), *On the infrastructure of the principal ideal class of an algebraic number field of unit rank one*, Mathematics of Computation, 50(182):569–579.
13. F. Fontein and P. Wocjan, (2011), *Quantum algorithm for computing the period lattice of an infrastructure*, arXiv:1111.1348.
14. F. Fontein, (2008), *Groups from cyclic infrastructures and Pohlig-Hellman in certain infrastructures*, arXiv:0803.2132.
15. F. Fontein, (2009), *The infrastructures of a global field of arbitrary unit rank*, arXiv:0809.1685.
16. F. Fontein, (2009), *Infrastructures and global fields*, http://math.fontein.de/infrastructures/.
17. J. W. Sands, (1991), *Generalization of a theorem by Siegel*, Acta Arithmetica, 58(1):47–57.
18. V. Arvind and P. Kurur Piyush, (2004) *On the complexity of computing units in a number field*, In Algorithmic Number Theory, Proc. of 6th International Symposium, ANTS-VI, Burlington, VT, June 2004, number 3076 in Lecture Notes on Computer Science, pages 72–86.
19. C. Thiel, (1995), *Short proofs using compact representations of algebraic integers*, Journal of Complexity, 11:310–329.
20. F. Fontein, (2011), Personal communication.
21. R. Schoof, (2008), *Computing Arakelov class groups*, Algorithmic Number Theory, MSRI Publications, Vol. 44, pages 447–495,
22. P. Kaye, R. Laflamme, and M. Mosca, (2007), *An introduction to quantum computing*. Oxford University Press.
23. S. R. Finch, (2003), *Mathematical constants*. Cambridge University Press.
24. D. Burton, (2010), *Elementary Number Theory*. McGraw-Hill, 7th edition.

Appendix A

We prove here some auxiliary results.

Lemma A.1: Let a and b be two random numbers chosen uniformly at random from $\{1,\ldots,N\}$. The probability that a and b are coprime is bounded from below by $1/2$, i.e.,

$$\Pr(\gcd(a,b)=1) > \frac{1}{2}\,. \tag{A.1}$$

Proof: Let p be an arbitrary prime. Then the probability that p divides a, denoted $\Pr(p \mid a)$, is given by

$$\Pr(p \mid a) = \frac{\lfloor \frac{N}{p} \rfloor}{N} \leq \frac{1}{p}\,.$$

Thus,

$$\Pr(p \mid \gcd(a,b)) \leq \frac{1}{p^2}\,.$$

We obtain an upper bound on the probability that there is a prime dividing the greatest common divisor of a and b with the help of the union bound. This yields

$$\Pr(\gcd(a,b) > 1) \leq \sum_p \frac{1}{p^2}\,,$$

where the summation index p ranges over all primes. The sum of squared reciprocals of primes is known to be

$$\sum_p \frac{1}{p^2} = \sum_{k=1}^{\infty} \frac{\mu(k)}{k} \ln \zeta(2k) = 0.4522474200\ldots\,,$$

where μ denotes the Möbius mu function and ζ the Riemann zeta function [23, page 95]. Finally, we obtain the desired result

$$\Pr(\gcd(a,b) = 1) \geq 1 - \sum_p \frac{1}{p^2} > \frac{1}{2}$$

by considering the complementary event $\square$.

We now prove a result related to continued fractions. The reader can find more details about continued fractions in [24].

Lemma A.2: Let p_i/q_i denote the convergents of a real number $r \in \mathbb{R}$, for $i \in \mathbb{N}$. Then for any given constant $c > 1$, there exists a convergent p_ℓ/q_ℓ such that $|r - p_\ell/q_\ell| < 1/cq_\ell$ and $q_\ell \leq c$.

Proof: Since $c > 1 = q_0$ and q_i form a monotonically increasing sequence for $i > 1$, there exists such a convergent p_ℓ/q_ℓ such that $q_\ell \leq c < q_{\ell+1}$ unless r has a finite continued fraction expansion with all the $q_i < c$. If the latter case occurs, then it follows that there exists a convergent p_ℓ/q_ℓ such that $r = p_\ell/q_\ell$ therefore for this convergent $|r - p/q| = 0 < 1/c$ and the statement of the lemma holds. Otherwise, r has a continued fraction expansion such that $q_\ell \leq c < q_{\ell+1}$. We know that the convergents satisfy the relation

$$\left| r - \frac{p_i}{q_i} \right| < \frac{1}{q_i q_{i+1}}.$$

Therefore, we must have

$$\left| r - \frac{p_\ell}{q_\ell} \right| < \frac{1}{q_\ell q_{\ell+1}} < \frac{1}{cq_\ell},$$

where we used the fact that $q_{\ell+1} > c$ $\square$.