From Quantum Mechanics to the Hidden Subgroup Problem: an Overview from Finite to Infinite Groups

Federico Di Concilio, Ramón Flores[0000−0002−5537−3299], Delaram Kahrobaei[0000−0001−5467−7832], and Marialaura Noce[0000−0002−3729−8598]

Abstract The Hidden Subgroup Problem (HSP for short) is an important problem in quantum computing and computational group theory, which has connections with several quantum algorithms, including Shor's algorithm for integer factorization and the discrete logarithm problem. The HSP can be stated as follows: given a function on a group that is constant and distinct on cosets of a hidden subgroup, determine the hidden subgroup efficiently. The problem captures the essence of exploiting group structures for computational advantage and provides insights into the power of quantum algorithms over classical approaches. While abelian instances of HSP are well-understood and efficiently solvable using the quantum Fourier transform, non-abelian cases remain challenging and are active areas of research. The goal of this survey is to put together all the basic background needed to understand well the Hidden Subgroup Problem in groups, reviewing recent results and discussing open problems.

Federico Di Concilio
University of Salerno
e-mail: fediconcilio@unisa.it

Ramón Flores
University of Seville
e-mail: cluje28@gmail.com

Delaram Kahrobaei
CUNY Graduate Center, PhD Program in Mathematics, CUNY Queens College, Departments of Computer Science and Mathematics, New York, United States, University of York, UK
e-mail: delaram.kahrobaei@qc.cuny.edu

Marialaura Noce
University of Salerno
e-mail: mnoce@unisa.it

1 Introduction

The National Security Agency (NSA) in 2015 revealed its intention to update security guidelines, aiming to replace all existing cryptographic systems with ones that are resistant to quantum attacks. To accomplish this, a new security standard must be approved by the National Institute of Standards and Technology (NIST). Various candidates for quantum-resistant algorithms and protocols have been submitted for evaluation in this standardization effort. The main families of post-quantum schemes considered in the NIST process are lattice-based, code-based, isogeny-based, multivariate-based, and hash-based; group-based cryptography is a further family studied in the post-quantum literature.

In particular, group-based cryptography attempts to leverage algorithmically hard tasks in group theory as the basis for secure primitives. Historically, most deployed public-key systems (RSA, discrete logarithm over finite fields and elliptic curves) rely on abelian group structures and are vulnerable to polynomial-time quantum attacks (e.g., Shor's algorithm [39]). Group-based cryptography explores non-abelian and combinatorial group families in search of problems that resist both classical and quantum attacks [24, 25]. Some classes that have recently been exploited include Engel groups [27], right-angled Artin groups [9–11], polycyclic groups [15], metabelian groups [16], pro p-groups [29], simple groups [13], some special linear groups [34], and automata groups [14, 26, 28].

The Hidden Subgroup Problem (HSP) is central to both quantum algorithm design and computational group theory.

Given a group G and an oracle for a function $f : G \to S$ that is constant and distinct on cosets of an unknown subgroup H, the HSP asks to recover H efficiently. This concise description underlies a family of quantum algorithms that lie at the intersection of combinatorial group theory and quantum mechanics. Formally, let G be a finite group and let $H \leq G$. A function $f : G \to S$ is said to hide H if for all $x, y \in G$ we have $f(x) = f(y)$ if and only if $x^{-1}y \in H$; equivalently, f is constant on left cosets of H and distinct on different cosets. The HSP asks: given oracle access to such an f, determine a generating set for H (or otherwise identify H) with high probability.

For example, in the abelian HSP case, the quantum Fourier transform reveals character information concentrated on the orthogonal complement of H, and classical postprocessing of independent samples reconstructs the subgroup in polynomial time. Simon's algorithm and Shor's factoring and discrete-logarithm algorithms are paradigmatic instances where abelian HSP techniques translate directly into powerful quantum procedures.

On the other hand, non-abelian HSPs expose deeper representation-theoretic structure: irreducible representations are matrix-valued, and coset structure imprints itself across representation labels and matrix entries. Measurement design and information extraction become substantially more delicate. Approaches explored in the literature include strong and weak Fourier sampling, and entangled measurements across multiple coset states. Each technique yields partial success for particular group families but falls short of a general solution for many groups of interest (for instance, the symmetric group and wide classes of matrix groups).

Beyond algorithmic interest, the HSP is very important in the post-quantum cryptography setting. Many classical cryptosystems rely on hardness assumptions in abelian groups and are therefore vulnerable to Shor attacks; proposals based on non-abelian or combinatorial group families aim to avoid these reductions.

Because many quantum algorithms correspond to HSP solutions, progress in non-abelian HSP algorithms could endanger cryptosystems whose hardness reduces to such subgroup-finding problems [4, 8, 35].

An overview of quantum algorithms with applications to group theory was given in [2]. In 2018, Horan and Kahrobaei [20] presented the state of the art of the HSP in groups and its connections with group-based cryptography. Recently, Dutto et al. surveyed the Hidden Subgroup Problem from a mathematical and cryptographic perspective [6].

This survey collects the complete background needed to understand HSPs in groups. We provide the mathematical notions of groups and representation theory, quantum algorithmic tools (state preparation, QFTs, measurement strategies), a list of solvable and hard HSP instances, and a discussion of their cryptographic relevance.

Organization

This paper is organized as follows. In Section 2 we provide a foundation in Hilbert spaces, including Dirac notation, operators, and tensor products. Section 3 introduces essential concepts from quantum mechanics, such as postulates, quantum registers, and state interpretation, to establish the computational context. Section 4 explains the basics of representation theory of finite groups, with applications to the Fourier transform. Putting everything together, in Section 5 we explain quantum gates and circuits, and in Section 6 we illustrate how classical computations can be made reversible and how quantum oracles function; key quantum algorithms, including the Deutsch, Deutsch-Jozsa, Simon and Shor algorithms, are examined to demonstrate practical applications of HSP principles. Then, in Sections 7-9, we present a detailed discussion of the HSP itself, covering its definition, solutions for finite abelian groups, and extensions to non-abelian and infinite groups. Finally, Section 10 collects open problems in the field. The survey is self-contained and it provides the necessary mathematical and physical background to fully understand the Hidden Subgroup Problem in groups.

2 Hilbert Spaces

In this section we introduce the algebraic notation used in quantum mechanics and recall basic facts about complex vector spaces. All vector spaces considered are finite dimensional.

2.1 Inner products and Dirac notation

We work with vector spaces over the field of complex numbers, and in particular with Hilbert spaces. Before introducing Dirac notation we recall the definition of an inner product and give an important example.

Definition 1 Let V be a complex vector space. An *inner product* on V is a map that assigns to each ordered pair of vectors $v, w \in V$ a complex number, denoted $\langle v, w\rangle$, and satisfying:

1. Linearity in the second argument: for all $v, w_1, w_2 \in V$ and $\alpha, \beta \in \mathbb{C}$,

$$\langle v, \alpha w_1 + \beta w_2\rangle = \alpha\langle v, w_1\rangle + \beta\langle v, w_2\rangle.$$

2. Conjugate symmetry: for all $v, w \in V$,

$$\langle v, w\rangle = \overline{\langle w, v\rangle}.$$

3. Positive-definiteness: for all $v \in V$, $\langle v, v\rangle \geq 0$, with equality iff $v = 0$.

Given an inner product $\langle\cdot,\cdot\rangle$ on V we define the norm of $v \in V$ by $\|v\| = \sqrt{\langle v, v\rangle}$. The distance induced by the inner product is $d(v, w) = \|v - w\|$.

Definition 2 A complex vector space V equipped with an inner product is a *Hilbert space* if it is complete with respect to the metric induced by the inner product. We denote Hilbert spaces by $\mathcal{H}$.

A standard example is $\mathbb{C}^n$ with the usual dot product.

Example 1 Let $\mathcal{H} = \mathbb{C}^n$. For column vectors $v = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$ and $w = \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix}$ the inner product is

$$\langle v, w\rangle = \sum_{i=1}^{n} v_i^* w_i,$$

where * denotes complex conjugation. With this inner product $\mathbb{C}^n$ is a Hilbert space.

Dirac notation is commonly used in quantum computing. A vector $v \in \mathcal{H}$ is written as a ket $|v\rangle$.

Example 2 Consider $\mathcal{H} = \mathbb{C}^2$ with canonical basis $e_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, e_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. In Dirac notation we write $e_1 = |0\rangle$ and $e_2 = |1\rangle$. Thus a vector $v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$ is written $|v\rangle = v_1|0\rangle + v_2|1\rangle$.

For n qubits the state space is $\mathbb{C}^{2^n}$. The canonical basis is labelled by binary strings of length n,

$$|00\ldots0\rangle,\ |00\ldots1\rangle,\ \ldots,\ |11\ldots1\rangle,$$

giving 2^n basis vectors.

The inner product is written as $\langle \cdot | \cdot \rangle$ in Dirac notation.

Remark 1 Every finite dimensional vector space with an inner product is automatically complete, and hence a Hilbert space.

We next define the dual space. For each ket $|v\rangle \in \mathcal{H}$ there is a corresponding bra $\langle v|$, which is a linear functional on $\mathcal{H}$.

Definition 3 The dual space $\mathcal{H}^*$ is the set of all linear maps from $\mathcal{H}$ to $\mathbb{C}$.

For $|v\rangle \in \mathcal{H}$ define $\langle v| : \mathcal{H} \to \mathbb{C}$ by $\langle v| (|w\rangle) = \langle v|w\rangle$ for all $|w\rangle \in \mathcal{H}$. In $\mathbb{C}^n$ the bra corresponding to $|v\rangle = (v_1, \ldots, v_n)^T$ is the row vector $\langle v| = (v_1^*, \ldots, v_n^*)$, i.e. obtain $\langle v|$ by taking the conjugate transpose of $|v\rangle$.

Example 3 In $\mathcal{H} = \mathbb{C}^2$, if

$$|\psi\rangle = \begin{pmatrix} 2-i \\ 1+2i \end{pmatrix},$$

then

$$\langle \psi| = \begin{pmatrix} 2+i & 1-2i \end{pmatrix}.$$

Some useful definitions follow.

Definition 4 Let $\mathcal{H}$ be a Hilbert space and let $\{|b_m\rangle\}_{m=1}^n$ be a basis.

1. Vectors $|v\rangle$ and $|w\rangle$ are *orthogonal* if $\langle v|w\rangle = 0$.
2. A vector $|v\rangle$ is a *unit vector* if $\| |v\rangle \| = 1$.
3. The basis $\{|b_m\rangle\}$ is *orthonormal* if $\langle b_i|b_j\rangle = \delta_{ij}$, where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

Example 4 In $\mathbb{C}^2$ the canonical basis $\{|0\rangle, |1\rangle\}$ is orthonormal. Another important orthonormal basis is the Hadamard basis:

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \qquad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

If $|v\rangle = \sum_i \alpha_i |b_i\rangle$ with $\{|b_i\rangle\}$ orthonormal, then $\| |v\rangle \|^2 = \sum_i |\alpha_i|^2$.

2.2 Operators and the outer product

Definition 5 A *linear operator* T on a Hilbert space $\mathcal{H}$ is a linear map $T : \mathcal{H} \to \mathcal{H}$.

Given $|v\rangle, |w\rangle \in \mathcal{H}$ the outer product $|v\rangle \langle w|$ is the linear operator on $\mathcal{H}$ defined by

$$(|v\rangle \langle w|) |h\rangle = \langle w|h\rangle \ |v\rangle \qquad (\forall |h\rangle \in \mathcal{H}).$$

Example 5 In $\mathcal{H} = \mathbb{C}^2$, for $|v\rangle = (v_1, v_2)^T$ and $|w\rangle = (w_1, w_2)^T$ the matrix of $|v\rangle \langle w|$ in the canonical basis is

$$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} (w_1^*, w_2^*) = \begin{pmatrix} v_1 w_1^* & v_1 w_2^* \\ v_2 w_1^* & v_2 w_2^* \end{pmatrix}.$$

In general, in $\mathbb{C}^n$ the outer product has matrix entries $v_i w_j^*$.

The set $\mathcal{L}(\mathcal{H})$ of all linear operators on $\mathcal{H}$ is a vector space. If $\{|b_i\rangle\}$ is an orthonormal basis of $\mathcal{H}$ then

$$\{|b_i\rangle \langle b_j| : i, j = 1, \ldots, n\}$$

is a basis of $\mathcal{L}(\mathcal{H})$. Any operator $T \in \mathcal{L}(\mathcal{H})$ can be written

$$T = \sum_{i,j} T_{ij} |b_i\rangle \langle b_j|, \qquad T_{ij} = \langle b_i| T |b_j\rangle .$$

The identity operator I satisfies $I = \sum_i |b_i\rangle \langle b_i|$.

Example 6 Important single-qubit operators are:

1. Pauli X: $X(\alpha |0\rangle + \beta |1\rangle) = \alpha |1\rangle + \beta |0\rangle$, represented by the matrix $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, and $X = |0\rangle \langle 1| + |1\rangle \langle 0|$.
2. Pauli Z: $Z(\alpha |0\rangle + \beta |1\rangle) = \alpha |0\rangle - \beta |1\rangle$, represented by $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, and $Z = |0\rangle \langle 0| - |1\rangle \langle 1|$.
3. Pauli Y: $Y(\alpha |0\rangle + \beta |1\rangle) = i\alpha |1\rangle - i\beta |0\rangle$, represented by $\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$, and $Y = -i |0\rangle \langle 1| + i |1\rangle \langle 0|$.
4. Hadamard H: $H(\alpha |0\rangle + \beta |1\rangle) = \frac{1}{\sqrt{2}}\big(\alpha(|0\rangle + |1\rangle) + \beta(|0\rangle - |1\rangle)\big)$, represented by $\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, and $H = \frac{1}{\sqrt{2}}(|0\rangle \langle 0| + |0\rangle \langle 1| + |1\rangle \langle 0| - |1\rangle \langle 1|)$.

We now define adjoints, unitaries, Hermitian operators and projectors.

Definition 6 Let T be a linear operator on $\mathcal{H}$. The *adjoint* $T^\dagger$ is the unique operator satisfying

$$\langle v| T^\dagger |w\rangle = \overline{\langle w| T |v\rangle}$$

for all $|v\rangle, |w\rangle \in \mathcal{H}$.

In matrix terms, if A represents T in the canonical basis of $\mathbb{C}^n$, then $A^\dagger = (A^*)^T$ is the conjugate transpose of A. For example, if

$$A = \begin{pmatrix} 1+i & 2+3i \\ 1 & 2-i \end{pmatrix},$$

then

$$A^\dagger = \begin{pmatrix} 1-i & 1 \\ 2-3i & 2+i \end{pmatrix}.$$

Definition 7 An operator U is *unitary* if $U^{-1} = U^\dagger$.

Unitary operators preserve inner products and norms.

Definition 8 An operator H is *Hermitian* (or self-adjoint) if $H = H^\dagger$.

Definition 9 A *projector* is an operator P with $P^2 = P$. An *orthogonal projector* additionally satisfies $P = P^\dagger$.

Eigenvalues are central in measurement theory: $|v\rangle$ is an eigenvector of T with eigenvalue λ if $T|v\rangle = \lambda|v\rangle$. Eigenvalues of Hermitian operators are real. Normal operators satisfy $TT^\dagger = T^\dagger T$; the spectral theorem states that every normal operator admits an orthonormal eigenbasis, so that

$$T = \sum_i \lambda_i |\phi_i\rangle \langle\phi_i| ,$$

where $\{|\phi_i\rangle\}$ are eigenvectors and $\{\lambda_i\}$ the corresponding eigenvalues (the spectrum of T).

2.3 Tensor products

Given Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ with bases $\{|b_i\rangle\}_{i=1}^n$ and $\{|c_j\rangle\}_{j=1}^m$, the tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$ has basis

$$\{|b_i\rangle \otimes |c_j\rangle : (i,j) \in \{1,\ldots,n\} \times \{1,\ldots,m\}\}.$$

A general element of $\mathcal{H}_1 \otimes \mathcal{H}_2$ is

$$|v\rangle = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_{ij} \ |b_i\rangle \otimes |c_j\rangle .$$

The tensor product satisfies the usual bilinearity properties:

1. $z(|v\rangle \otimes |w\rangle) = (z|v\rangle) \otimes |w\rangle = |v\rangle \otimes (z|w\rangle)$ for all $z \in \mathbb{C}$.
2. $(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = |v_1\rangle \otimes |w\rangle + |v_2\rangle \otimes |w\rangle$.
3. $|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = |v\rangle \otimes |w_1\rangle + |v\rangle \otimes |w_2\rangle$.

For operators $T : \mathcal{H}_1 \to \mathcal{H}_1$ and $W : \mathcal{H}_2 \to \mathcal{H}_2$ the tensor product operator $T \otimes W$ acts by

$$(T \otimes W)(|v\rangle \otimes |w\rangle) = (T|v\rangle) \otimes (W|w\rangle),$$

and extends linearly to the whole space.

3 Fundamentals of quantum mechanics

This section outlines the mathematical framework of quantum mechanics as it applies to systems used in quantum computing. We introduce the qubit, the basic unit of quantum information, and summarize the core postulates needed for quantum computation.

3.1 Postulates

The link between the mathematical formalism of quantum theory and experimental facts is expressed through a set of postulates.

State Space Postulate An isolated quantum system is associated with a Hilbert space $\mathcal{H}$. The physical pure states correspond to unit vectors in $\mathcal{H}$. For the systems we consider, the elementary building block is a two-level system: a qubit. Its state space is $\mathcal{H} = \mathbb{C}^2$. We choose an orthonormal basis denoted by $\{|0\rangle, |1\rangle\}$ called the computational basis.
In general a qubit is represented by

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,,$$

where α, β are such that $|\alpha|^2 + |\beta|^2 = 1$. Unlike a classical bit, which is either 0 or 1, a qubit can be in coherent superpositions of $|0\rangle$ and $|1\rangle$.

Evolution Postulate The evolution of a closed quantum system between two instants is described by a unitary operator U on the system's Hilbert space. If the system is initially in $|\psi_1\rangle$ then after the evolution it is in $|\psi_2\rangle = U\,|\psi_1\rangle$. In quantum computation, unitary operators on a single qubit are called single-qubit gates and are represented by 2×2 unitary matrices. For instance, the Pauli X operator (quantum NOT) is

$$X = |0\rangle\,\langle 1| + |1\rangle\,\langle 0| = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

which maps $|0\rangle$ to $|1\rangle$.

Composition Postulate The state space of a composite system is the tensor product of the component state spaces. If subsystem A is in $|\psi_A\rangle$ and subsystem B in $|\psi_B\rangle$ then the joint state is $|\psi_A\rangle \otimes |\psi_B\rangle$ (often written $|\psi_A\psi_B\rangle$). For n qubits the joint space is $(\mathbb{C}^2)^{\otimes n}$ of dimension 2^n.
Not every state of a composite system can be factored: when it cannot be written as a tensor product of single-system states the subsystems are entangled. For example, the two-qubit state

$$|\Phi\rangle = \frac{1}{\sqrt{2}}\big(|01\rangle - |10\rangle\big)$$

is entangled and cannot be expressed as $|\phi_1\rangle \otimes |\phi_2\rangle$.
A widely used set of maximally entangled two-qubit states (Bell states) is

$$|\Phi^{\pm}\rangle = \frac{|00\rangle \pm |11\rangle}{\sqrt{2}}, \qquad |\Psi^{\pm}\rangle = \frac{|01\rangle \pm |10\rangle}{\sqrt{2}},$$

which form an orthonormal basis of $\mathbb{C}^2 \otimes \mathbb{C}^2$.

Measurement Postulate Projective (von Neumann) measurements correspond to an orthogonal decomposition of the identity $I = \sum_i P_i$ into projectors P_i. If the system is in

$$|\psi\rangle = \sum_i \alpha_i |\psi_i\rangle$$

with $\{|\psi_i\rangle\}$ an orthonormal basis, then a measurement in that basis yields outcome i with probability $p(i) = |\alpha_i|^2$ and the post-measurement state is $|\psi_i\rangle$. More generally, for projectors P_i the probability of outcome i is $p(i) = \langle\psi| P_i |\psi\rangle$ and the post-measurement state (when $p(i) > 0$) is $P_i |\psi\rangle / \sqrt{p(i)}$.

For a single-qubit state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ a measurement in the computational basis yields 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$.

As an example, consider the two-qubit state

$$|\chi\rangle = \sqrt{\tfrac{3}{14}} |00\rangle + \sqrt{\tfrac{4}{14}} |01\rangle + \sqrt{\tfrac{5}{14}} |10\rangle + \sqrt{\tfrac{2}{14}} |11\rangle .$$

A computational-basis measurement returns 00 with probability $3/14$, 01 with $4/14$, 10 with $5/14$, and 11 with $2/14$.

As another example, let

$$|\xi\rangle = \sqrt{\tfrac{1}{6}} |0010\rangle + \sqrt{\tfrac{1}{3}} |0101\rangle + \sqrt{\tfrac{1}{2}} |0011\rangle .$$

If we measure the second qubit (counting from the left) in the computational basis, the outcome 0 occurs with probability $1/6 + 1/2 = 2/3$, and the post-measurement (renormalized) state becomes

$$\sqrt{\tfrac{1}{4}} |0010\rangle + \sqrt{\tfrac{3}{4}} |0011\rangle ,$$

while the outcome 1 occurs with probability $1/3$ and collapses the state to $|0101\rangle$.

Projective measurements can also be described via observables: any Hermitian operator $M = \sum_i m_i P_i$ has a spectral decomposition whose projectors P_i define a measurement returning eigenvalue m_i with probability $\langle\psi| P_i |\psi\rangle$.

3.2 Quantum registers

An n-qubit quantum register is a unit vector in a 2^n-dimensional Hilbert space with computational basis states

$$|x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle ,$$

where each $x_i \in \{0, 1\}$. We usually abbreviate this product as $|x_1 x_2 \cdots x_n\rangle$.

For instance, the two-qubit computational basis is $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ which correspond to the standard basis vectors in $\mathbb{C}^4$. A general two-qubit register is

$$|\psi\rangle = \alpha_1 |00\rangle + \alpha_2 |01\rangle + \alpha_3 |10\rangle + \alpha_4 |11\rangle,$$

with $\sum_{i=1}^{4} |\alpha_i|^2 = 1$.

The state space dimension grows exponentially with n (dimension 2^n), which underlies the large state-space resources available to quantum algorithms.

3.3 Physical interpretation

A qubit can be physically realized by any two well-separated energy levels or two orthogonal degrees of freedom. Common physical implementations include:

- photon polarizations (horizontal and vertical),
- a spin-$\frac{1}{2}$ particle in a magnetic field (spin-up and spin-down),
- two discrete electronic energy levels of an atom.

4 Finite groups and discrete Fourier transform

In this section, we define the discrete Fourier transform on finite groups. To do this, we will use the theory of representations of finite groups, in particular character theory.

We begin our analysis by recalling the standard definition of the discrete Fourier transform.

Definition 10 The discrete Fourier transform, transforms a sequence of N complex numbers $x_0, \ldots, x_{N-1}$, into another sequence of complex numbers $X_0, \ldots, X_{N-1}$, which is defined by:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i \frac{k}{N} n},$$

for $k = 0, \ldots, N-1$.

The goal of this section is to establish a link between group theory and Fourier analysis. Each sequence $x_0, \ldots, x_{N-1}$ of complex numbers can be associated with a function $f : \mathbb{Z}_N \to \mathbb{C}$ such that $f([k]_N) = x_k$, for $k = 0, \ldots, N-1$, and vice versa. As is well known, the set $L(\mathbb{Z}_N)$, of functions $f : \mathbb{Z}_N \to \mathbb{C}$, is a vector space over the complex field $\mathbb{C}$, and therefore the discrete Fourier transform is a linear operator on the vector space $L(\mathbb{Z}_N)$. We can state that the standard discrete Fourier transform is the Fourier transform defined on the cyclic group $\mathbb{Z}_N$. Now the question is how to define the discrete Fourier transform on any group? As discussed above, to define the Fourier transform on any group, we will use the theory of representations of finite groups. In

the following, we denote the set of functions $f : G \to \mathbb{C}$, with the symbol $L(G)$. The set $L(G)$ is a Hilbert space with inner product defined by

$$\langle f, g \rangle = \frac{1}{|G|} \sum_{x \in G} f(x)^* g(x)$$

for every $f, g \in L(G)$.

In the following, we will first define the standard Fourier transform using the character theory of a finite group. Next, we will define the Fourier transform for abelian groups, and finally define the transform for arbitrary finite groups.

We start with the definition of representation of a finite group on a finite dimensional vector space, the main idea of this definition is that the elements of a finite group can be regarded as invertible matrices.

Definition 11 Let G be a finite group and let V be a vector space over $\mathbb{C}$ of dimension m.

A representation of G on the vector space V is a group homomorphism

$$\varphi : G \to GL(m, \mathbb{C}),$$

where $GL(m, \mathbb{C})$ is the group of $m \times m$ invertible matrices over $\mathbb{C}$.

So to each element g of the group G we are associating, through the homomorphism φ, an invertible matrix φ_g whose elements are complex numbers. We observe that for a finite group G there can exist different representations, for example, if the group $G = \mathbb{Z}_n$, we can define the trivial representation

$$\phi : g \in \mathbb{Z}_n \to 1 \in \mathbb{C}^* = GL(1, \mathbb{C}).$$

Apart from that, we can also define two other representations α and β of $\mathbb{Z}_n$. For example

$$\alpha : \mathbb{Z}_n \to GL(2, \mathbb{C}),$$

such that

$$\alpha_{\overline{m}} = \begin{bmatrix} \cos(\frac{2\pi m}{n}) & -\sin(\frac{2\pi m}{n}) \\ \sin(\frac{2\pi m}{n}) & \cos(\frac{2\pi m}{n}) \end{bmatrix}$$

which is the matrix for rotation by $\frac{2\pi m}{n}$, and

$$\beta : \mathbb{Z}_n \to GL(2, \mathbb{C}),$$

such that

$$\beta_{\overline{m}} = \begin{bmatrix} e^{\frac{2\pi m i}{n}} & 0 \\ 0 & e^{-\frac{2\pi m i}{n}} \end{bmatrix}.$$

The representations α and β satisfy the following property, which tells us that, even if they have a different form, they are actually the same representation up to an invertible matrix, that is, there exists

$$A = \begin{bmatrix} i & -i \\ 1 & 1 \end{bmatrix},$$

a 2×2 invertible matrix such that:

$$A^{-1}\alpha_{\overline{m}}A = \beta_{\overline{m}},$$

for every $\overline{m} \in \mathbb{Z}_n$.

This leads us to the important notion of equivalence.

Definition 12 Two representations $\varphi : G \to GL(n, \mathbb{C})$ and $\psi : G \to GL(n, \mathbb{C})$ are equivalent if there exists an invertible matrix A such that

$$A^{-1}\varphi_g A = \psi_g,$$

for every $g \in G$.

An important subgroup of $GL(n, \mathbb{C})$ is the group of unitary matrices

$$U(n, \mathbb{C}) = \{A \in GL(n, \mathbb{C}) : A^{-1} = A^{\dagger}\},$$

where the matrix $A^{\dagger}$ is the conjugate transpose of matrix A.

Definition 13 A representation $\varphi : G \to GL(n, \mathbb{C})$ of a finite group G is a *unitary representation* if

$$\varphi_g \in U(n, \mathbb{C}),$$

for every $g \in G$, i.e., each element g of the finite group G is associated with a unitary matrix A.

Unitary representations play a fundamental role in the theory of representations of finite groups, as it can be proved that every representation is equivalent to a unitary representation, so from what we said previously, if we want to study the representations of a finite group it is sufficient to study the unitary representations.

Definition 14 Let $\varphi : G \to GL(n, \mathbb{C})$ be a representation of G on the vector space $V = \mathbb{C}^n$.

- A subspace W of the vector space V is said to be G-invariant if $\varphi_g w \in W$ for every $g \in G$ and $w \in W$.
- A representation $\varphi : G \to GL(n, \mathbb{C})$ is said to be irreducible if the only G-invariant subspaces of V are $\{0\}$ and V.

A key theorem in this field is the following,

Theorem 1 *Let G be a finite group, then there are finitely many irreducible unitary representations up to equivalence.*

To each representation of a group we can associate a function from the group to the field of complex numbers, this function is called character and encodes different information about the representation.

Definition 15 Let $\varphi : G \to GL(n, \mathbb{C})$ be a representation of the finite group G. The character of the representation φ is the function

$$\chi : G \to \mathbb{C}$$

such that

$$\chi(g) = Tr(\varphi_g),$$

where $Tr(A)$ is the trace of the matrix A, i.e. the sum of the elements on the diagonal of A. The character associated with an irreducible representation is called an irreducible character.

If the group G is abelian of order N, then G has N unitary irreducible representations $\varphi_0, \ldots, \varphi_{N-1}$ up to equivalence, let $\chi_0, \ldots, \chi_{N-1}$ be the characters of these representations, for example the group $\mathbb{Z}_N$ has the following N irreducible characters, $\chi_k : \mathbb{Z}_N \to \mathbb{C}$ such that

$$\chi_k([m]_N) = e^{\frac{2\pi i k m}{N}},$$

for $k = 0, \ldots, N-1$.

Furthermore, if G is an abelian group, the dimension of the vector space $L(G)$ is equal to the cardinality of G and its orthonormal basis is the set made up of the irreducible characters of G. Therefore every function $f : G \to \mathbb{C}$ is such that

$$f = \langle \chi_0, f \rangle \chi_0 + \ldots + \langle \chi_{N-1}, f \rangle \chi_{N-1}.$$

In particular, every function

$$f : \mathbb{Z}_N \to \mathbb{C},$$

is such that

$$f([m]_N) = \sum_{k=0}^{N-1} \langle \chi_k, f \rangle e^{\frac{2\pi i k m}{N}}, \qquad m = 0, \ldots, N-1.$$

The standard discrete Fourier transform can be defined using character theory by observing that

$$N\langle \chi_k, f \rangle = \sum_{n=0}^{N-1} x_n e^{-2\pi i \frac{k}{N} n},$$

where $x_n = f([n]_N)$, for $n = 0, \ldots, N-1$. We are now in a position to provide the definition of a Fourier transform for abelian groups.

Definition 16 Let $G = \{g_1, \ldots, g_N\}$ be an abelian group of order N, and let $\chi_1, \ldots, \chi_N$ be the irreducible characters of G (indexed through a fixed bijection between G and its set of irreducible characters). The Fourier transform $T : L(G) \to L(G)$ is such that

$$T(f)(g_k) = N\langle \chi_k, f\rangle,$$

for $k = 1, \ldots, N$.

For an arbitrary finite group G, let $\widehat{G}$ denote a complete set of pairwise inequivalent irreducible unitary representations of G, and let d_ρ be the dimension of $\rho \in \widehat{G}$.

Definition 17 Let G be a finite group. The *Fourier transform* of $f \in L(G)$ at $\rho \in \widehat{G}$ is the $d_\rho \times d_\rho$ matrix

$$\widehat{f}(\rho) = \sqrt{\frac{d_\rho}{|G|}} \sum_{g \in G} f(g)\, \rho(g).$$

Since $\sum_{\rho \in \widehat{G}} d_\rho^2 = |G|$, the map $f \mapsto (\widehat{f}(\rho))_{\rho \in \widehat{G}}$ is a linear isomorphism between $L(G)$ and $\bigoplus_{\rho \in \widehat{G}} \mathbb{C}^{d_\rho \times d_\rho}$. When G is abelian, all the d_ρ are equal to 1, and we recover the previous definition up to normalization and complex conjugation of the characters. The *quantum Fourier transform* over G is the unitary operator

$$|g\rangle \mapsto \sum_{\rho \in \widehat{G}} \sqrt{\frac{d_\rho}{|G|}} \sum_{i,j=1}^{d_\rho} \rho(g)_{ij}\ |\rho, i, j\rangle\ .$$

Measuring only the label ρ of the resulting state is called *weak Fourier sampling*, while measuring also the row and column indices i, j (in some chosen basis) is called *strong Fourier sampling*.

5 Quantum gates and circuits

The dynamical behaviour of a quantum register is controlled by unitary transformations. In the language of computation, these transformations are organized into quantum gates, which are elementary, reversible operations acting on one or more qubits. By arranging gates in a prescribed order we obtain a quantum circuit, the analogue of a classical logic network. The goal of this section is to introduce the standard single- and two-qubit gates and to explain how circuits built from them model quantum algorithms.

5.1 Single- and two-qubit gates

Quantum computation proceeds by applying unitary gates to qubits.

The single-qubit Pauli X (quantum NOT) is

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

which transforms $\alpha\,|0\rangle + \beta\,|1\rangle$ into $\beta\,|0\rangle + \alpha\,|1\rangle$.

Other important single-qubit gates include

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \qquad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

where H maps $|0\rangle$ to $(|0\rangle + |1\rangle)/\sqrt{2}$, producing an equal superposition of computational basis states.

Two-qubit gates can create entanglement. The controlled-NOT (CNOT) gate acts on control qubit A and target qubit B by sending

$$|A, B\rangle \mapsto |A, B \oplus A\rangle\,.$$

On the computational basis:

$$|00\rangle \mapsto |00\rangle\,, \quad |01\rangle \mapsto |01\rangle\,, \quad |10\rangle \mapsto |11\rangle\,, \quad |11\rangle \mapsto |10\rangle\,,$$

and its matrix in the ordered basis $\{|00\rangle\,, |01\rangle\,, |10\rangle\,, |11\rangle\}$ is

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

A controlled-U gate generalizes this construction: when the control is $|1\rangle$ the unitary U is applied to the target; when the control is $|0\rangle$ the target is unchanged. The quantum Fourier transform is an important multi-qubit unitary defined by

$$\mathrm{QFT}_N\,|x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i x y/N}\,|y\rangle\,,$$

for $x \in \{0, \ldots, N-1\}$. Note that, compared with Definition 10, the QFT uses the opposite sign in the exponent and the normalization $1/\sqrt{N}$, which makes it unitary.

5.2 Quantum circuits

Quantum circuits are sequences of quantum gates acting on registers. For example, the three-CNOT construction swaps two qubits: applying CNOT with a control and b target, then CNOT with b control and a target, then CNOT with a control and b target exchanges the states of a and b.

Any unitary on n qubits may be implemented (up to desired precision) by composing a suitable sequence of elementary gates drawn from a universal gate set, often including single-qubit rotations and a two-qubit entangling gate such as CNOT.

5.2.1 Quantum complexity

In analogy with classical computational complexity, one can introduce complexity classes arising from quantum circuits. Below we present the definition of the class BQP, while the classical class BPP is only described informally. Roughly speaking, BQP consists of all decision problems that can be solved by a quantum computer in polynomial time with bounded error. Analogously, the corresponding classical class, BPP, contains those problems that can be solved in polynomial time by a probabilistic algorithm whose probability of error is bounded. It is known that

$$\mathsf{BPP} \subseteq \mathsf{BQP},$$

although it remains an open question whether this inclusion is strict.

Definition 18 A language $L \subseteq \{0,1\}^*$ belongs to BQP (Bounded-Error Quantum Polynomial Time) if there exists a deterministic Turing machine running in polynomial time that, on input 1^n, outputs a description of a quantum circuit C_n satisfying:

- C_n takes n input qubits and, in addition, uses a polynomial number of fixed $|0\rangle$ input;
- C_n has a single designated primary output qubit;
- When the primary output qubit of C_n is measured on input $|w\rangle|0\cdots 0\rangle$, the output v obeys

$$\Pr[v = 1 \text{ iff } w \in L] \geq \frac{2}{3}.$$

Equivalently, $(C_n)_n$ is a polynomial-time uniform family of quantum circuits with error probability at most $\frac{1}{3}$.

It is still not known how BQP and NP are related.

6 Quantum algorithms

In this section we will describe some of the early quantum algorithms. These algorithms are simple and illustrate the main ingredients behind the more useful and powerful quantum algorithms we describe in the subsequent sections.

All the algorithms we will see are based on the fact that each function $f : \{0,1\}^n \to \{0,1\}^m$ can be viewed as a quantum gate U_f. As we have already seen, quantum gates are unitary operators and thus in particular are invertible functions. Therefore, we want to tackle the problem of whether one can replace each function $f : \{0,1\}^n \to \{0,1\}^m$ by an invertible function $\overline{f}$. In the next section, we will try to answer this question.

6.1 Making Irreversible Gates Reversible

We start this section with the following definitions.

Definition 19 A classical gate is a function $f : \{0,1\}^n \to \{0,1\}^m$, with $n, m \in \mathbb{N}$.

Definition 20 A classical gate $f : \{0,1\}^n \to \{0,1\}^m$ is reversible if it is an invertible function. If f is not invertible, we will say that f is an irreversible gate.

Example 7 An example of a reversible gate is the NOT gate, that is, the function

$$NOT : \{0,1\} \to \{0,1\},$$

such that $NOT(0) = 1, NOT(1) = 0$. The function

$$AND : \{0,1\}^2 \to \{0,1\},$$

such that $AND(a,b) = ab$, is an irreversible gate.

Note that if

$$f : \{0,1\}^n \to \{0,1\}^m$$

is such that $n \neq m$, then f is an irreversible gate.

One might wonder how to realize a reversible counterpart to the classical AND operation. Any candidate for a reversible AND must be defined on a domain whose size matches that of its codomain. In other words, one cannot realize a reversible AND as a function from $\{0,1\}^2$ to $\{0,1\}$. A reversible implementation therefore requires enlarging the output space so that the input and output sets have the same cardinality.

We therefore consider the function $f : \{0,1\}^3 \to \{0,1\}^3$ such that $f(a,b,c) = (a,b,ab \oplus c)$. It is easy to prove that f is an invertible function, hence a reversible gate, moreover f encodes all the information of the AND function, in fact if $c = 0$, then $ab \oplus c = ab = AND(a,b)$. This remark works in general, not only for the AND gate, suppose we have an irreversible gate $f : \{0,1\}^2 \to \{0,1\}$, we can define $\overline{f} : \{0,1\}^3 \to \{0,1\}^3$ such that $\overline{f}(a,b,c) = (a,b,f(a,b) \oplus c)$. The gate $\overline{f}$ is a reversible gate that encodes all the information of the f gate. We can generalise this to any irreversible gate, let us consider another example before considering the general case, let $f : \{0,1\}^2 \to \{0,1\}^2$ be an irreversible gate such that $f(a,b) = (f_1(a,b), f_2(a,b))$, we can define the gate $\overline{f} : \{0,1\}^4 \to \{0,1\}^4$ such that $\overline{f}(a,b,c,d) = (a,b,f_1(a,b) \oplus c, f_2(a,b) \oplus d)$. The gate $\overline{f}$ is a reversible gate that encodes all the information of the f gate. In general, for every irreversible gate $f : \{0,1\}^n \to \{0,1\}^m$, such that $f(\mathbf{v}) = (f_1(\mathbf{v}), \ldots, f_m(\mathbf{v}))$, we can define

$$\overline{f} : \{0,1\}^{n+m} \to \{0,1\}^{n+m},$$

such that

$$\overline{f}(\mathbf{w}) = (\mathbf{v}, f_1(\mathbf{v}) \oplus w_{n+1}, \ldots, f_m(\mathbf{v}) \oplus w_{n+m}),$$

where w_i is the i-th entry of the $\mathbf{w}$ vector, and $\mathbf{v} = (w_1, \ldots, w_n)$. Therefore, for every irreversible gate, there is a reversible gate to replace it.

6.2 Quantum Oracles

A classical oracle is simply a Boolean function, meaning a function that acts on bits. So an oracle can be seen as a classical gate. In the following algorithms, we will always start with a classical oracle that will be replaced by a quantum oracle. A quantum oracle is simply a unitary operator, so a quantum oracle is, in particular, a bijective function. In order to replace a classical oracle with a quantum oracle, the classical oracle must necessarily be invertible. We have seen that every classical gate, and thus every classical oracle, can be replaced with a reversible gate. It can therefore be assumed that every oracle is invertible. For every classical oracle

$$f : \{0,1\}^n \to \{0,1\}^m,$$

we can define the quantum oracle

$$U_f : \mathbb{C}^{2^n} \otimes \mathbb{C}^{2^m} \to \mathbb{C}^{2^n} \otimes \mathbb{C}^{2^m},$$

such that $U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle$, for each $x \in \{0,1\}^n$ and for each $y \in \{0,1\}^m$.

6.3 The Deutsch Algorithm

Let us now look at our first quantum algorithm. Deutsch's algorithm is a very simple example of a quantum algorithm, it is a good starting point because, although very simple and easy to understand, Deutsch's algorithm illustrates the key ideas of quantum parallelism, which are used in all quantum algorithms. The problem solved by Deutsch's algorithm is as follows. Suppose we have a reversible circuit to compute an unknown function $f : \{0,1\} \to \{0,1\}$. We treat this reversible circuit as a "black box" or "oracle". This means that we can apply the circuit to obtain information about the value of $f(x)$ for a given input x, but not to obtain information about the internal mechanism that governs the circuit. The problem is to determine if the function f is "constant" or "balanced" by making queries to f.

Definition 21 A function $f : \{0,1\} \to \{0,1\}$ is constant if $f(x) = f(y)$ for every $x, y \in \{0,1\}$. A function $f : \{0,1\} \to \{0,1\}$ is balanced if it is not constant.

It is clear that, in a classical context, two queries must be made to the oracle to determine whether the function f is constant or balanced. Deutsch's algorithm is a quantum algorithm capable of solving this problem by making a single query to a quantum oracle for f. In general, any classical reversible circuit can be transformed into a quantum circuit by replacing each classical reversible gate of the given circuit with the analogous quantum gate. In particular, in our case, the reversible circuit used to compute f can be replaced by the analogous quantum circuit. This quantum circuit can be expressed as a unitary operator $U_f |xy\rangle = |x\rangle |y \oplus f(x)\rangle$. Having created a quantum version of the circuit for f, one can provide quantum bits as inputs. The algorithm is as follows:

1. $|\psi_1\rangle = |0\rangle \otimes |1\rangle$;
2. $|\psi_2\rangle = H^{\otimes 2} |\psi_1\rangle = \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle)$;
3. $|\psi_3\rangle = U_f |\psi_2\rangle = \frac{1}{2}(|0\rangle |f(0) \oplus 0\rangle - |0\rangle |f(0) \oplus 1\rangle + |1\rangle |f(1) \oplus 0\rangle - |1\rangle |f(1) \oplus 1\rangle)$;
4. $|\psi_4\rangle = (H \otimes I) |\psi_3\rangle$, and measure the first qubit.

Let us divide the reasoning into two parts, suppose that f is constant and that $f(1) = f(0) = 0$, then

$$|\psi_3\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}},$$

$$|\psi_4\rangle = (H \otimes I) |\psi_3\rangle = |0\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

By measuring $|\psi_4\rangle$ in the first qubit, we obtain $|0\rangle$ with probability 1.

Let us now assume f constant and $f(0) = f(1) = 1$, then

$$|\psi_3\rangle = -\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}},$$

$$|\psi_4\rangle = (H \otimes I) |\psi_3\rangle = -|0\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

By measuring $|\psi_4\rangle$ in the first qubit, we obtain $|0\rangle$ with probability 1.

Assuming that f is balanced, it can be shown that

$$|\psi_3\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}},$$

or

$$|\psi_3\rangle = -\frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}},$$

in both cases if we measure $|\psi_4\rangle = (H \otimes I) |\psi_3\rangle$ in the first qubit, we obtain $|1\rangle$ with probability 1. We have thus shown that the function f is constant if and only if by measuring $|\psi_4\rangle$ we obtain $|0\rangle$ with probability 1, the function is balanced otherwise.

6.4 The Deutsch-Jozsa Algorithm

The Deutsch-Jozsa algorithm solves a problem that is a direct generalisation of the problem solved by the Deutsch algorithm. We are given a reversible circuit implementing an unknown function f, but this time f is a function from n-bit strings to a single bit. That is,

$$f : \{0,1\}^n \to \{0,1\}.$$

The function f is either constant (meaning $f(x)$ is the same for all x), or f is balanced (meaning $f(x) = 0$ for exactly half of the input strings x, and $f(x) = 1$ for the other half of the inputs). The problem is to determine whether f is constant, or balanced, by making queries to the circuit for f. In the worst case, using a classical algorithm we

cannot decide with certainty whether f is constant or balanced using less than $2^{n-1}+1$ queries, the Deutsch-Jozsa algorithm will determine whether f is constant or balanced by making a single query to a quantum version U_f of the reversible circuit for f. Before presenting the algorithm, we note that

$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} (-1)^{x\cdot z}|z\rangle,$$

for every $x \in \{0,1\}^n$ and $x \cdot z$ is the scalar product between x and z modulo 2.

The algorithm is as follows:

1. $|\psi_1\rangle = |0\rangle^n \otimes |1\rangle$;
2. $|\psi_2\rangle = H^{\otimes(n+1)}|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \frac{|0\rangle-|1\rangle}{\sqrt{2}}$;
3. $|\psi_3\rangle = U_f|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \frac{|f(x)\oplus 0\rangle-|f(x)\oplus 1\rangle}{\sqrt{2}}$;

Note that if $f(x) = 0$, then

$$\frac{|f(x)\oplus 0\rangle - |f(x)\oplus 1\rangle}{\sqrt{2}} = \frac{|0\rangle - |1\rangle}{\sqrt{2}},$$

if $f(x) = 1$, then

$$\frac{|f(x)\oplus 0\rangle - |f(x)\oplus 1\rangle}{\sqrt{2}} = \frac{|1\rangle - |0\rangle}{\sqrt{2}},$$

therefore

$$\frac{|f(x)\oplus 0\rangle - |f(x)\oplus 1\rangle}{\sqrt{2}} = (-1)^{f(x)}\frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

We can rewrite

$$|\psi_3\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)}|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

To obtain $|\psi_4\rangle$, we apply $H^{\otimes n}$ to the first register,

$$|\psi_4\rangle = \frac{1}{2^n} \sum_{x,y\in\{0,1\}^n} (-1)^{f(x)+x\cdot y}|y\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

The amplitude of $|0\rangle^n$ in the first register is $\frac{1}{2^n}\sum_x (-1)^{f(x)}$. Hence, measuring the first register, f is constant if and only if the outcome 0^n occurs with probability 1, and f is balanced if and only if it occurs with probability 0.

6.5 Simon's Algorithm

Let f be a function that takes as input a binary string of length n and outputs a binary string of length n, so $f : \{0,1\}^n \to \{0,1\}^n$. The promise is that there exists a string $s \neq 0^n$, of n bits, such that $f(x) = f(y)$ if and only if $y \in \{x, x \oplus s\}$.

Example 8 Assume that $n = 3$, $s = 110$ and that $f(000) = f(110) = 011$, $f(001) = f(111) = 101$, $f(010) = f(100) = 001$, $f(011) = f(101) = 000$.

The problem is to determine the string s by making queries to the oracle f. Simon's problem requires an exponential number of queries on a classical computer. We assume that we have the following reversible black-box for implementing f, $U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle$. Initially, we have two n-qubit registers, one for the input qubits, and another for the answer qubits,

$$|\psi_1\rangle = |0^n\rangle |0^n\rangle ,$$

now we apply the Hadamard gate to each of input qubits, resulting in

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |0^n\rangle ,$$

$$|\psi_3\rangle = U_f |\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle .$$

Now, we again apply the Hadamard gate to each of the input qubits,

$$|\psi_4\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} (-1)^{x \cdot z} |z\rangle |f(x)\rangle .$$

Now, let us measure the answer qubits. We will get one particular value, f', of $f(x)$, by hypothesis there are x', x'' such that $f(x') = f(x'') = f'$. Therefore the state collapses to these two values:

$$|\psi_5\rangle = \frac{1}{\sqrt{2}} \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} [(-1)^{x' \cdot z} + (-1)^{x'' \cdot z}] |z\rangle |f'\rangle .$$

Note that $(-1)^{x' \cdot z} + (-1)^{x'' \cdot z} \in \{0, 2, -2\}$, when measuring the input qubits, we only get a value of $|z\rangle$ where $(-1)^{x' \cdot z} + (-1)^{x'' \cdot z} \in \{2, -2\}$, that is $x' \cdot z = x'' \cdot z \bmod 2$. Therefore, it follows that $x' \cdot z + x'' \cdot z = 0 \bmod 2$, and $(x' + x'') \cdot z = s \cdot z = 0 \bmod 2$. So, when measuring the input qubits, we get a value of $|z\rangle$ such that its dot product with s is 0 mod 2. Since the vectors z obtained in this way lie in the subspace $s^{\perp} = \{z : z \cdot s = 0 \bmod 2\}$, which has dimension $n - 1$, repeating the quantum algorithm $O(n)$ times we obtain, with high probability, $n - 1$ linearly independent values $z_1, \ldots, z_{n-1}$. The homogeneous linear system $z_i \cdot s = 0 \bmod 2$, $i = 1, \ldots, n-1$, then has exactly two solutions, 0^n and s, and s is the nonzero one.

Example 9 Consider the function f defined in Example 8, we apply the algorithm of Simon to determine $s = 110$. So,

$$|\psi_1\rangle = |0^3\rangle\,|0^3\rangle\,,$$

we apply the Hadamard gate to each of input qubits,

$$|\psi_2\rangle = \frac{1}{\sqrt{2^3}} \sum_{x\in\{0,1\}^3} |x\rangle\,|000\rangle = \frac{1}{\sqrt{2^3}}[|000\rangle + |001\rangle + \ldots + |111\rangle]\,|000\rangle\,,$$

therefore,

$$|\psi_2\rangle = \frac{1}{\sqrt{2^3}}[|000\rangle\,|000\rangle + \ldots + |111\rangle\,|000\rangle].$$

Now apply U_f to $|\psi_2\rangle$, that is

$$|\psi_3\rangle = \frac{1}{\sqrt{2^3}}[|000\rangle\,|011\rangle + \ldots + |111\rangle\,|101\rangle].$$

We measure the quantum register $|\psi_3\rangle$ in $|f(x)\rangle = |011\rangle$, we obtain the state

$$|\psi_4\rangle = \frac{1}{\sqrt{2}}[|000\rangle + |110\rangle]\,|011\rangle\,,$$

apply the Hadamard gate to each of the input qubits,

$$|\psi_5\rangle = \frac{1}{4} \sum_{z\in\{0,1\}^3} \left[(-1)^{000\cdot z}+(-1)^{110\cdot z}\right]|z\rangle\,|011\rangle = \frac{1}{2}\big(|000\rangle+|001\rangle+|110\rangle+|111\rangle\big)\,|011\rangle\,,$$

e.g. measuring 001, $|\psi_5\rangle$ collapses into $|z_1\rangle = |001\rangle$. In order to determine the value of s, we must repeat the algorithm until we obtain two (that is, $n-1$) linearly independent values of $|z\rangle$. Then, suppose we repeat the algorithm and obtain as the value of $|z_2\rangle = |111\rangle$, we can accept this value of $|z_2\rangle$, because $|z_1\rangle$ and $|z_2\rangle$ are linearly independent. Now suppose we obtain $|z_3\rangle = |110\rangle$. Solving the system $z_1 \cdot s = 0$, $z_2 \cdot s = 0$, that is $s_3 = 0$ and $s_1 + s_2 + s_3 = 0 \bmod 2$, we obtain $s \in \{000, 110\}$, and hence $s = 110$. Note that a further outcome such as $|z_3\rangle = |110\rangle = |z_1 \oplus z_2\rangle$ would give no new information.

6.6 Shor's Algorithms

6.6.1 Shor's Factoring Algorithm

Shor's algorithm is based on a fundamental result from number theory: the function

$$F(a) = x^a \bmod n$$

is periodic when x is an integer coprime with n. In Shor's algorithm, n is the number to be factored. Two integers are said to be coprime if their greatest common divisor is equal to 1. On a classical computer, evaluating this function for an exponential number of values of a would require exponential time. Shor's algorithm overcomes this limitation by exploiting quantum parallelism, allowing all these evaluations to be performed simultaneously. The usefulness of this function for factoring large integers lies in its periodicity: since $F(a)$ has a period r, we have $x^0 \bmod n = 1$, and $x^r \bmod n = 1$, $x^{2r} \bmod n = 1$, and so on. Using this information, we can perform the following algebraic manipulations. Starting from

$$x^r \equiv 1 \pmod{n},$$

we obtain

$$(x^{r/2})^2 \equiv 1 \pmod{n},$$

which implies

$$(x^{r/2})^2 - 1 \equiv 0 \pmod{n}.$$

If the period r is even, this expression can be factored as

$$(x^{r/2} - 1)(x^{r/2} + 1) \equiv 0 \pmod{n}.$$

This shows that the product $(x^{r/2} - 1)(x^{r/2} + 1)$ is a multiple of n, the number to be factored. Provided that $x^{r/2} \not\equiv -1 \pmod{n}$ (note that $x^{r/2} \not\equiv 1 \pmod{n}$, since r is the order of x), at least one of the factors $x^{r/2} - 1$ or $x^{r/2} + 1$ must share a nontrivial common divisor with n. Therefore, by computing

$$\gcd(x^{r/2} - 1, n) \quad \text{and} \quad \gcd(x^{r/2} + 1, n),$$

where gcd denotes the greatest common divisor, we can obtain a nontrivial factor of n. Shor's algorithm for factoring an integer n can be divided into several steps:

1. **Check simple cases:** Determine whether n is a prime number, an even number, or a perfect power $n = a^b$ with $b \geq 2$. If n falls into any of these categories, Shor's algorithm is not needed, since efficient classical methods exist to identify these cases and provide factors if applicable. This step is performed on a classical computer.
2. **Choose q:** Select an integer q that is a power of 2 such that

$$n^2 \leq q < 2n^2.$$

   This step is also carried out classically.
3. **Select a coprime integer x:** Pick a random integer x that is coprime with n, meaning $\gcd(x, n) = 1$. Efficient classical methods exist for finding such an x. This step is performed on a classical computer.
4. **Initialize quantum registers:** Prepare a quantum register divided into two parts, register 1 and register 2, so that the total state can be written as $|\text{reg1}, \text{reg2}\rangle$. Register 1 must have enough qubits to represent integers up to $q - 1$, while register 2 must

be able to represent integers up to $n-1$. The calculation of the required number of qubits is done classically.

5. **Prepare superposition:** Load register 1 with an equal superposition of all integers from 0 to $q-1$, and initialize register 2 to all zeros. This operation is performed by the quantum computer. At this stage, the quantum memory register is in the state:

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a, 0\rangle.$$

6. **Compute modular exponentiation:** Apply the transformation $x^a \bmod n$ to each number stored in register 1 and store the result in register 2. Thanks to quantum parallelism, a single application of a reversible circuit for modular exponentiation, of size polynomial in $\log n$, evaluates $x^a \bmod n$ on all the basis states of the superposition simultaneously. After this step, the register is in the state:

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a, x^a \bmod n\rangle.$$

7. **Measure the second register:** Observe some value k in register 2. This measurement collapses the register 1 into an equal superposition of all integers a such that

$$x^a \bmod n = k.$$

Denote the set of such integers by A, and let $|A|$ be the number of elements in A. The state of the quantum register after measurement is:

$$\frac{1}{\sqrt{|A|}} \sum_{a' \in A} |a', k\rangle.$$

8. **Apply the discrete Fourier transform:** Perform the discrete Fourier transform (DFT) on register 1. For a state $|a\rangle$, the DFT maps it to

$$|a\rangle \mapsto \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} |c\rangle \, e^{2\pi i a c/q}.$$

This step is executed by the quantum computer and is crucial for extracting the period of the function $F(a) = x^a \bmod n$, which will ultimately allow us to factor n. The QFT over $\mathbb{Z}_q$ can be implemented with $O((\log q)^2)$ elementary gates. After the DFT, the register is in the state:

$$\frac{1}{\sqrt{|A|}} \sum_{a' \in A} \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} |c, k\rangle \, e^{2\pi i a' c/q}.$$

9. **Measure register 1:** Measure the state of register 1 and call the observed value m. This integer is, with high probability, close to a multiple of q/r, where r is the period we aim to find. This measurement is performed by the quantum computer.
10. **Classical post-processing:** Use the measured value m to compute the period r on a classical computer. In particular:
    - The value m is likely to satisfy $|m/q - \lambda/r| \leq 1/(2q)$, where λ is an integer.
    - By performing the division m/q as a floating-point number, one can find the best rational approximation of m/q whose denominator is less than n (via the continued fraction expansion of m/q). This gives an estimate of r.
    - We take this denominator as a candidate for r.
    - If $x^r \not\equiv 1 \pmod{n}$, small multiples of the candidate can be tested; if the order r turns out to be odd, the procedure is restarted with a new x. Efficient classical methods exist for this post-processing.
11. **Compute factors of n:** Once r has been determined, a factor of n can be obtained by computing
$$\gcd(x^{r/2} - 1, n) \quad \text{and} \quad \gcd(x^{r/2} + 1, n).$$
    If a nontrivial factor is found, the algorithm terminates. Otherwise, one returns to step 3 and repeats the process. This step is performed on a classical computer.
12. **Handling failures:** Step 11 also accounts for possible failures of Shor's algorithm. For example:
    - The measurement in step 9 could be 0, making post-processing impossible.
    - The algorithm might find trivial factors such as 1 or n, which are not useful.

    Therefore, the algorithm includes a provision to return to step 3 and retry with a new random x if necessary.

7 The Hidden Subgroup Problem

In this section, we recall the definition of the Hidden Subgroup Problem (HSP), and we present a systematic explanation in the case in which the group G is abelian. This problem in abelian groups can be solved with a polynomial time algorithm on a quantum computer, and the solution relies on a "natural" standard quantum approach that is based on the quantum Fourier transform. For example, Shor's algorithm for integer factorization can be seen as an instance of the HSP on cyclic groups.

Definition 22 Let $f : G \to X$ be a function that maps a group G to a finite set X. This function has the property that there exists a subgroup H of G such that for any $x, y \in G$, it holds that $f(x) = f(y)$ if and only if $xH = yH$. The objective of the Hidden Subgroup Problem (HSP) is to find a generating set for the subgroup H.

In this case, such an f is a hiding function, and H is a hidden subgroup. Classically, the function f is given by either a subroutine or an oracle that returns some values $f(x)$. Thus, given a group G and a hiding function f, the HSP asks to find a generating set for H.

7.1 The hidden subgroup problem for finite abelian groups

In this section, we discuss the hidden subgroup problem for finite abelian groups. Note that one of the first results in this direction, after Simon's algorithm for $\mathbb{Z}_2^n$, was proved by Shor in 1994, where he provided an algorithm using the Fourier transform for groups of the form $\mathbb{Z}_N$ and $\mathbb{Z}_N \times \mathbb{Z}_N$, for a positive integer N. The year after, Kitaev described a polynomial-time quantum algorithm for the abelian stabilizer problem, generalizing Shor's celebrated results. His approach relies on a technique for estimating the eigenvalues of a unitary operator. As a further application, this technique yields a polynomial-time quantum algorithm for computing the Fourier transform over any finite abelian group, [30]. In 2022 Imran and Ivanyos gave an exact quantum polynomial-time algorithm for $\mathbb{Z}_{m^h}^k$. The algorithm is based on the quantum Fourier transform modulo m and does not assume that a factorization of m is known. They further describe applications in the case of solvable groups [21].

7.2 Step-by-step description of the finite abelian case

Let N and k be positive integers, and let S be an arbitrary subgroup of $\mathbb{Z}_N^k$. We associate to S the quantum state given by the uniform superposition over its elements,

$$|S\rangle = \frac{1}{\sqrt{|S|}} \sum_{s \in S} |s\rangle \,,$$

where $|S|$ denotes the cardinality of the subgroup. This state represents an equal-amplitude superposition of all elements of S.

We define the orthogonal (or dual) subgroup of S as

$$S^{\perp} = \{t \in \mathbb{Z}_N^k \mid t \cdot s = 0 \text{ for all } s \in S\},$$

where the inner product $t \cdot s$ is computed modulo N. The subgroup $S^{\perp}$ therefore consists of all elements of $\mathbb{Z}_N^k$ that are orthogonal to every element of S with respect to this inner product.

A fundamental property of the quantum Fourier transform is that applying the k-fold quantum Fourier transform $QFT_N^{\otimes k}$ to the state $|S\rangle$ yields a uniform superposition over the elements of the orthogonal subgroup,

$$QFT_N^{\otimes k} |S\rangle = \sum_{t \in S^{\perp}} |t\rangle \,,$$

up to normalization factors. This result shows that the quantum Fourier transform maps subgroup states to states supported on their orthogonal complements.

For any fixed element $b \in \mathbb{Z}_N^k$, we define the coset state corresponding to the translate $b + S$ as

$$|b+S\rangle = \frac{1}{\sqrt{|S|}} \sum_{s\in S} |b+s\rangle\,.$$

This state is a shifted version of $|S\rangle$ and represents a uniform superposition over the elements of the coset $b+S$.

Applying the quantum Fourier transform to this coset state gives

$$QFT_N^{\otimes k}\,|b+S\rangle = \frac{1}{\sqrt{|S^\perp|}} \sum_{t\in S^\perp} e^{\frac{2\pi i}{N}\,t\cdot b}\,|t\rangle\,.$$

Compared to the unshifted case, a phase factor depending on b appears, encoding the translation information in the Fourier domain.

This framework generalizes naturally to arbitrary finite abelian groups of the form

$$G = \mathbb{Z}_{N_1} \times \cdots \times \mathbb{Z}_{N_k},$$

where $N_1, \ldots, N_k$ are positive integers. For any subgroup $S \subseteq G$, we define the associated subgroup and coset states as

$$|S\rangle = \frac{1}{\sqrt{|S|}} \sum_{s\in S} |s\rangle\,, \qquad |b+S\rangle = \frac{1}{\sqrt{|S|}} \sum_{s\in S} |b+s\rangle\,.$$

In this more general setting, the orthogonal subgroup $S^\perp$ is defined as the set of all elements $t \in G$ satisfying

$$\frac{t_1 s_1}{N_1} + \cdots + \frac{t_k s_k}{N_k} = 0 \mod 1 \quad \text{for all } s \in S,$$

where the condition $x = 0 \bmod 1$ means that $x \in \mathbb{Z}$. Equivalently, this condition can be expressed as

$$e^{2\pi i\left(\frac{t_1 s_1}{N_1}+\cdots+\frac{t_k s_k}{N_k}\right)} = 1 \quad \text{for all } s \in S,$$

i.e., the corresponding character evaluates trivially on S.

Finally, let $N = |G| = N_1 \cdots N_k$, and let $N = p_1^{n_1} p_2^{n_2} \cdots p_\ell^{n_\ell}$ denote its prime factorization, and define

$$n = \sum_{j=1}^{\ell} n_j.$$

For simplicity, we assume throughout that the quantum Fourier transform over $\mathbb{Z}_N$, denoted by QFT_N, can be implemented exactly and efficiently for all values of N. This assumption allows us to treat Fourier transforms over finite abelian groups arising from such decompositions without introducing additional technical complications.

We now describe an algorithm for the finite abelian Hidden Subgroup Problem (HSP). Let $\mathcal{H}_X$ denote the Hilbert space corresponding to the output register of the oracle U_f.

7.2.1 Algorithm for the finite abelian HSP

1. Initialize the counter by setting $i = 1$.
2. Prepare the initial state
$$|0\rangle|0\rangle \cdots |0\rangle|0\rangle \in \mathcal{H}_{N_1} \otimes \mathcal{H}_{N_2} \otimes \cdots \otimes \mathcal{H}_{N_k} \otimes \mathcal{H}_X.$$
3. Apply the quantum Fourier transform $\mathrm{QFT}_{N_1,N_2,\ldots,N_k}$ to the input registers.
4. Apply the unitary operator U_f to obtain the state
$$\frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle|f(x)\rangle.$$
5. (Optional) Measure the second register.
6. Apply the inverse quantum Fourier transform $\mathrm{QFT}^{-1}_{N_1,N_2,\ldots,N_k}$ to the input registers.
7. Measure the first register, yielding a value t_i.
8. If $i < n + 4$, increment i and repeat the procedure from Step 2; otherwise, continue to the next step.
9. Compute generators $k_1, k_2, \ldots$ of the solution space of the equation
$$Tx^T = 0 \pmod 1,$$
where T is the matrix whose i-th row is given by
$$\left(\frac{t_{i,1}}{N_1}, \frac{t_{i,2}}{N_2}, \ldots, \frac{t_{i,k}}{N_k}\right).$$
10. Output the generators $k_1, k_2, \ldots$.

8 The Hidden Subgroup Problem in groups

The HSP has been extensively investigated in several other families of groups, with results that range from efficient quantum algorithms to strong evidence of intractability. In this section, we focus on some recent results on the HSP in some families of groups.

8.1 The Hidden Subgroup Problem for dihedral groups

Let $D_n = \langle r, s \mid r^n = s^2 = 1, srs = r^{-1} \rangle$ be the dihedral group of order $2n$. For dihedral groups, one of the most interesting and studied cases is when the hidden subgroup H is a subgroup of order 2, i.e.:
$$H = \langle r^k s \rangle \quad \text{for some } k \in \{0, 1, \ldots, n-1\}.$$

The first results for the HSP are due to Ettinger and Høyer in 2000 [7]. After that, the work of Kuperberg [31] provides a subexponential-time algorithm using a collision technique with quantum Fourier transform. Regev gave a variant of Kuperberg's algorithm that uses only polynomial space, at the cost of a slightly larger (still subexponential) running time [38]. Recently in 2022 Fujita proposed a simple quantum algorithm for the dihedral HSP (DHSP) over dihedral groups of order a power of two, which runs in polynomial time under heuristic assumptions [12]. On the other hand, in 2024 Chen and Sun, apart from giving a complete overview of the known results related to the HSP in dihedral groups, prove new results regarding the hardness assumptions of the HSP [3]. More precisely, they exploit the dihedral coset problem (DCP) and its connections with cloning quantum states.

8.2 The Hidden Subgroup Problem and semidirect products

The Semidirect Discrete Logarithm Problem (SDLP for short) was first studied in 2013 in [17]. In 2024 in [1], the authors present new developments and for the first time a detailed and precise analysis of the SDLP. The authors in this work use group exponentiation, that is products of the form

$$\phi^{x-1}(g) \cdot \ \ldots \ \cdot \phi(g) \cdot g,$$

where g is an element of a (semi)group, ϕ is an endomorphism and x is a positive integer. Given g, ϕ, and $h = \phi^{x-1}(g) \ldots \phi(g)g$, the SDLP asks whether it is possible to recover x. Products of this form have enough structure to be cryptographically useful, and SDLP was originally proposed as a problem with no known reduction to a Hidden Subgroup Problem. In [1] the authors show that SDLP reduces to an abelian hidden shift problem, which yields a subexponential quantum algorithm via Kuperberg's techniques.

8.3 The Hidden Subgroup Problem for symmetric groups

The HSP over the symmetric group S_n is of particular interest because the graph isomorphism problem can be reduced to it. However, strong negative results are known: Moore, Russell, and Schulman [35] proved that the standard approach based on strong Fourier sampling over S_n fails to solve the HSP, even with an exponential number of samples. Hallgren, Moore, Rötteler, Russell, and Sen [19] showed that entangled measurements across a small number of registers are also insufficient. These results suggest that fundamentally new techniques are needed for the HSP in S_n.

8.4 The Hidden Subgroup Problem for wreath products

Wreath products provide a natural family of groups interpolating between abelian and non-abelian settings. Ivanyos, Magniez, and Santha [22] gave efficient quantum algorithms for several instances of the non-abelian HSP: hidden normal subgroups of solvable and permutation groups, and arbitrary hidden subgroups of groups with a small commutator subgroup or admitting an elementary abelian normal 2-subgroup of small index or with cyclic quotient. The latter class includes the wreath products $\mathbb{Z}_2^k \wr \mathbb{Z}_2$. The HSP in wreath products of the form $\mathbb{Z}_2 \wr \mathbb{Z}_n$ is closely related to certain shift problems and remains an active area of research.

8.5 The Hidden Subgroup Problem for nilpotent groups and PSL$(2, q)$

Ivanyos, Sanselme, and Santha [23] showed that the HSP can be solved efficiently in nil-2 groups, that is, in groups of nilpotency class at most 2. Their algorithm also contains a classical reduction of the HSP in nilpotent groups of constant class to the case of p-groups of exponent p with a trivial or cyclic hidden subgroup. Radhakrishnan, Rötteler, and Sen [37] showed that Fourier sampling in random bases solves the HSP in the Heisenberg group of order p^3, a nil-2 group, with polynomially many samples. For the projective special linear group $\mathrm{PSL}(2, q)$, Denney, Moore, and Russell [5] investigated the structure of the HSP and its connections with problems in number theory. The case of $\mathrm{PSL}(2, q)$ is particularly relevant due to its simplicity as a finite group of Lie type and its connections with expander graphs and cryptographic protocols.

9 The Hidden Subgroup Problem for infinite groups

In this section, we will focus on one of the most recent topics in the setting of Hidden Subgroup Problem in groups.

The computational difficulty of the HSP varies significantly depending on the underlying group G. The most interesting instances of the HSP are those for which a quantum computer can solve the problem efficiently, while no efficient classical algorithm is known. When the function f is provided as an oracle, there is often an unconditional argument showing that the HSP cannot be solved efficiently using classical computation. The most familiar example arises from Shor's algorithm, which solves the HSP for $G = \mathbb{Z}$ and $H = r\mathbb{Z}$, where r is the period of $a \mapsto x^a \bmod n$ (see Section 6.6.1). Very recently, in 2025 Kuperberg explored the study of HSP in infinite (discrete) groups [33]. His main contributions are related to the alternative decision problem of the HSP with $G = \mathbb{Q}$, the rational numbers seen as a discrete group under addition.

More precisely, Kuperberg stated some results concerning hardness and algorithmic results.

To state the first hardness results, it is worth defining the hidden subgroup existence problem (HSEP for short). The HSEP problem is a decision problem which asks whether the subgroup H is non-trivial.

Theorem 2 ([33]) *The Hidden Subgroup Existence Problem in the quotient $\mathbb{Q}/\mathbb{Z}$ is NP-complete (under quantum polynomial-time reductions).*

The idea behind the proof is the following. Let $\mathcal{D}(x)$ be a decision problem in the class NP, and suppose that there exists a predicate $z(x, y)$ that accepts only prime numbers $y = p$ as certificates. We define a hidden subgroup $H \leq \mathbb{Q}$ generated by 1, together with $1/p$ for every prime p accepted by z.

Their approach is to construct an H-periodic hiding function $f(a/b)$, which takes a rational number a/b as input and can be efficiently evaluated using access to the predicate z. This construction relies on the fact that rational numbers admit a canonical partial fraction decomposition, analogous to the partial fraction decomposition of rational functions in calculus. For example, one can write

$$\frac{7}{30} = 1 - \frac{1}{2} - \frac{1}{3} + \frac{1}{5}.$$

Computing this partial fraction decomposition requires factoring the denominator b, which is why the reduction is in BQP. A classical counterpart exists using partial factorization algorithms, but it depends on certain conjectures in number theory.

The function $f(a/b)$ is defined by discarding the integer part and discarding all terms whose denominator p is prime and accepted by z. The subgroup H always contains $\mathbb{Z}$, and $H \neq \mathbb{Z}$ if and only if there exists an accepted certificate. Consequently, the decision problem $\mathcal{D}$ reduces to an instance of the hidden subgroup problem over $\mathbb{Q}$ (equivalently, to deciding whether the subgroup $H/\mathbb{Z}$ of $\mathbb{Q}/\mathbb{Z}$ is nontrivial).

The theorem above contrasts with both Shor's algorithm for $G = \mathbb{Z}$ and Hallgren's algorithm when $G = \mathbb{R}$, where the hiding function is Lipschitz and maps into a Hilbert space, [18]. Assuming the conjecture that NP $\not\subseteq$ BQP, the theorem above implies that the Hidden Subgroup Problem in $\mathbb{Q}$ has no efficient quantum algorithm.

Theorem 3 ([33]) *The Hidden Subgroup Existence Problem in a finitely generated non-abelian free group F_k is NP-complete, even under the assumption that the hidden subgroup is normal.*

On the algorithmic side, Kuperberg proved the following.

Theorem 4 ([33]) *The Hidden Subgroup Problem (HSP) in $\mathbb{Z}^k$, when vectors are represented in binary, can be solved efficiently on a quantum computer (i.e., in BQP), in time polynomial in both the dimension k and the bit size of the solution.*

This result contrasts with the Shor-Kitaev algorithm, which works only under the additional assumption that the hidden subgroup $H \leq \mathbb{Z}^k$ has full rank (i.e. $l = k$, where l is the rank of H), that is when H is a finite-index subgroup. The case where H has lower rank $l < k$ appears to be a new result.

Theorem 5 ([33]) *Let G be a finitely generated group that contains a finite-index abelian subgroup K. Then the Hidden Subgroup Problem in G can be solved in time $2^{O(\sqrt{n})}$, where n is the bit complexity of the solution.*

A special case of this theorem reduces to the hidden shift problem in the abelian subgroup K. Unlike the previous theorems, this result can be obtained using existing algorithms. Specifically, it follows from prior work on the dihedral hidden subgroup problem, and the proof can be simplified using a refinement of the second algorithm introduced by Kuperberg [32], as recently noted by Peikert [36].

10 Open problems

We conclude this chapter by listing some open problems related to the Hidden Subgroup Problem.

1. **HSP in the symmetric group.** The most prominent open problem is whether there exists an efficient quantum algorithm for the HSP in the symmetric group S_n. A positive solution would yield a quantum algorithm for the graph isomorphism problem. Despite significant effort, neither efficient quantum algorithms nor unconditional hardness results are known for this case.
2. **HSP in $\mathrm{GL}_n(\mathbb{F}_q)$ and finite groups of Lie type.** Determine the quantum complexity of the HSP for the general linear group $\mathrm{GL}_n(\mathbb{F}_q)$ and, more broadly, for finite groups of Lie type. While partial results exist for small cases and for specific families of subgroups, the general problem remains open.
3. **HSP in wreath products.** Characterize the complexity of the HSP for wreath products $A \wr B$, where A and B range over various families of groups. In particular, the case $\mathbb{Z}_2 \wr \mathbb{Z}_n$ (closely related to hidden shift problems) is not fully understood. Is there a polynomial-time quantum algorithm for the HSP in $\mathbb{Z}_p \wr \mathbb{Z}_p$ for general primes p?
4. **Beyond strong Fourier sampling.** Develop new quantum algorithmic techniques for the non-abelian HSP that go beyond the standard framework of strong Fourier sampling, which has been shown to be insufficient for both S_n and certain families of 2-groups. What additional quantum primitives or algebraic structures can be exploited?
5. **HSP in infinite non-abelian groups.** Following Kuperberg's work [33], investigate the HSP in other infinite groups, such as lattices in Lie groups, Baumslag-Solitar groups, or arithmetic groups. In particular, it is open whether the HSP in the discrete Heisenberg group $H_3(\mathbb{Z})$ can be solved efficiently.
6. **Connections with lattice problems.** Clarify the relationship between the HSP in dihedral groups D_n and lattice problems such as the unique shortest vector problem (uSVP). Although reductions in both directions are partially known, the precise complexity-theoretic equivalence remains open.

References

1. C. Battarbee, D. Kahrobaei, L. Perret, and S. F. Shahandashti. A subexponential quantum algorithm for the semidirect discrete logarithm problem. In M.-J. Saarinen and D. Smith-Tone, editors, *Post-Quantum Cryptography (PQCrypto 2024)*, pages 202–226. Springer, Cham, 2024.
2. M. Batty, S. L. Braunstein, A. J. Duncan, and S. Rees. Quantum algorithms in group theory. arXiv:quant-ph/0310133, 2003.
3. I. Chen and D. Sun. The dihedral hidden subgroup problem. *J. Math. Cryptol.*, 18(1):Paper No. 20220029, 2024.
4. A. M. Childs and W. van Dam. Quantum algorithms for algebraic problems. *Rev. Modern Phys.*, 82(1):1–51, 2010.
5. A. Denney, C. Moore, and A. Russell. Finding conjugate stabilizer subgroups in PSL and PSp: a generalization of the hidden subgroup problem. *Quantum Inf. Comput.*, 10(3&4):282–291, 2010.
6. S. Dutto, P. Mercuri, N. Murru, and L. Romano. A survey about Hidden Subgroup Problem from a mathematical and cryptographic perspective. arXiv:2512.02087, 2025.
7. M. Ettinger and P. Høyer. On quantum algorithms for noncommutative hidden subgroups. *Adv. in Appl. Math.*, 25(3):239–251, 2000.
8. M. Ettinger, P. Høyer, and E. Knill. The quantum query complexity of the hidden subgroup problem is polynomial. *Inform. Process. Lett.*, 91(1):43–48, 2004.
9. R. Flores, D. Kahrobaei, and T. Koberda. Algorithmic problems in right-angled Artin groups: complexity and applications. *J. Algebra*, 519:111–129, 2019.
10. R. Flores, D. Kahrobaei, and T. Koberda. An algebraic characterization of k-colorability. *Proc. Amer. Math. Soc.*, 149(5):2249–2255, 2021.
11. R. Flores, D. Kahrobaei, and T. Koberda. Hamiltonicity via cohomology of right-angled Artin groups. *Linear Algebra Appl.*, 631:94–110, 2021.
12. H. Fujita. A heuristic approach to the dihedral hidden subgroup problem. In *Future Computing 2022: The Fourteenth International Conference on Future Computational Technologies and Applications*, pages 1–2. IARIA, 2022.
13. M. I. González Vasco, D. Kahrobaei, and E. McKemmie. Applications of finite non-abelian simple groups to cryptography in the quantum era. *La Matematica*, 3(2):588–603, 2024.
14. R. I. Grigorchuk and D. Grigoriev. Key agreement based on automaton groups. *Groups Complex. Cryptol.*, 11(2):77–81, 2019.
15. J. Gryak and D. Kahrobaei. The status of polycyclic group-based cryptography: a survey and open problems. *Groups Complex. Cryptol.*, 8(2):171–186, 2016.
16. J. Gryak, D. Kahrobaei, and C. Martínez-Pérez. On the conjugacy problem in certain metabelian groups. *Glasg. Math. J.*, 61(2):251–269, 2019.
17. M. Habeeb, D. Kahrobaei, C. Koupparis, and V. Shpilrain. Public key exchange using semidirect product of (semi)groups. In M. Jacobson, M. Locasto, P. Mohassel, and R. Safavi-Naini, editors, *Applied Cryptography and Network Security (ACNS 2013)*, volume 7954 of *Lecture Notes in Computer Science*, pages 475–486. Springer, Heidelberg, 2013.
18. S. Hallgren. Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem. *J. ACM*, 54(1):Art. 4, 2007.
19. S. Hallgren, C. Moore, M. Rötteler, A. Russell, and P. Sen. Limitations of quantum coset states for graph isomorphism. *J. ACM*, 57(6):Art. 34, 2010. Preliminary version in STOC 2006.
20. K. Horan and D. Kahrobaei. The hidden subgroup problem and post-quantum group-based cryptography. In J. H. Davenport, M. Kauers, G. Labahn, and J. Urban, editors, *Mathematical Software - ICMS 2018*, volume 10931 of *Lecture Notes in Computer Science*, pages 218–226. Springer, Cham, 2018.
21. M. Imran and G. Ivanyos. An exact quantum hidden subgroup algorithm and applications to solvable groups. *Quantum Inf. Comput.*, 22(9&10):770–789, 2022.
22. G. Ivanyos, F. Magniez, and M. Santha. Efficient quantum algorithms for some instances of the non-abelian hidden subgroup problem. *Internat. J. Found. Comput. Sci.*, 14(5):723–739, 2003.
23. G. Ivanyos, L. Sanselme, and M. Santha. An efficient quantum algorithm for the hidden subgroup problem in nil-2 groups. *Algorithmica*, 62(1-2):480–498, 2012.

24. D. Kahrobaei, R. Flores, and M. Noce. Group-based cryptography in the quantum era. *Notices Amer. Math. Soc.*, 70(5):752–763, 2023.
25. D. Kahrobaei, R. Flores, M. Noce, M. E. Habeeb, and C. Battarbee. *Applications of group theory in cryptography: post-quantum group-based cryptography*, volume 278 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2024.
26. D. Kahrobaei, A. A. Malik, and D. Savchuk. Contracting self-similar groups in group-based cryptography. arXiv:2408.14355, 2024.
27. D. Kahrobaei and M. Noce. Algorithmic problems in Engel groups and cryptographic applications. *Int. J. Group Theory*, 9(4):231–250, 2020.
28. D. Kahrobaei, M. Noce, and E. Rodaro. Applications of automaton groups in cryptography. *Int. J. Comput. Math. Comput. Syst. Theory*, 9(2):96–106, 2024.
29. D. Kahrobaei and M. Stanojkovski. Cryptographic multilinear maps using pro-p groups. *Adv. Math. Commun.*, 17(5):1101–1114, 2023.
30. A. Y. Kitaev. Quantum measurements and the abelian stabilizer problem. arXiv:quant-ph/9511026, 1995.
31. G. Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM J. Comput.*, 35(1):170–188, 2005.
32. G. Kuperberg. Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. In *8th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2013)*, volume 22 of *LIPIcs*, pages 20–34. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2013. arXiv:1112.3333.
33. G. Kuperberg. The hidden subgroup problem for infinite groups. arXiv:2507.18499, 2025.
34. C. Le Coz, C. Battarbee, R. Flores, T. Koberda, and D. Kahrobaei. Post-quantum hash functions using $\mathrm{SL}_n(\mathbb{F}_p)$. *Adv. Math. Commun.*, 19(3):996–1009, 2025.
35. C. Moore, A. Russell, and L. J. Schulman. The symmetric group defies strong Fourier sampling. *SIAM J. Comput.*, 37(6):1842–1864, 2008.
36. C. Peikert. He gives C-sieves on the CSIDH. In *Advances in Cryptology – EUROCRYPT 2020*, volume 12106 of *Lecture Notes in Computer Science*, pages 463–492. Springer, 2020.
37. J. Radhakrishnan, M. Rötteler, and P. Sen. On the power of random bases in Fourier sampling: hidden subgroup problem in the Heisenberg group. In *Automata, Languages and Programming (ICALP 2005)*, volume 3580 of *Lecture Notes in Computer Science*, pages 1399–1411. Springer, 2005. arXiv:quant-ph/0503114.
38. O. Regev. A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space. arXiv:quant-ph/0406151, 2004.
39. P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 124–134. IEEE, 1994.