SIMULATING QUANTUM CIRCUITS WITH HARD-CORE BOSONS

David Emmanuel-Costa[a]

[a]Institute of Software Technology, German Aerospace Center (DLR), Cologne, Germany

ABSTRACT

Simulating quantum systems without the exponential overhead characteristic of classical simulations is a challenging task. Complex Clifford algebras can unify all components of a quantum circuit, with only elements of the algebra together with the geometric product. However, due to its non-Abelian nature, the geometric product realizes the tensor product by constantly correcting the sign of the expressions. In this work, hard-core bosons are shown as the most natural way to fully represent multiqubit systems and the tensor product is naturally realized without the need of any sign correction. It is proved that the two approaches are strictly equivalent up to a Jordan-Wigner transformation. Multiqubit gates in terms of annihilation and creation operators are described. Finally, a method to simulate a quantum circuit is developed.

Keywords: hard-core bosons, complex Clifford algebra, quantum circuit representation.

1 INTRODUCTION

During the last decades, quantum computing, as proposed initially by Feynman [1] (for older works connecting quantum mechanics with computation see Refs. [2, 3]) and later formalized by Deutsch [4], has emerged as a new, both theoretical and experimental, multidisciplinary field of science. It comprises many aspects of computer science, physics, and mathematics that utilizes quantum mechanics with the aim to solve complex problems faster than on classical computers. Quantum computers could lead to computational tasks be executed exponentially faster on a quantum processor than on a classical processor [1] - often referred as quantum supremacy [5]. A comprehensive recent review is available in Ref. [6]. Although, the modern quantum computation has reached the noisy intermediate-scale quantum (NISQ) era [7], one is still at early stages for transiting to a fault-tolerant era, which is essential to enable quantum advantage.

In the standard approach of quantum computation, the basic unit is the qubit, the analogue of the classical bit. It is defined by the vector state

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

built upon an orthonormal basis states $\{|0\rangle\,,|1\rangle\}$ of the two-dimensional complex Hilbert space, $\mathbb{C}^2$. The complex coefficients α, β are just restricted to the relation $|\alpha|^2+|\beta|^2=1$, so that $|\psi\rangle$ remains normalized to 1. A multiqubit system is then defined by taking the tensor product of replicas of the complex Hilbert space $\mathbb{C}^2$. A quantum computer made of n qubits corresponds to a complex Hilbert space of dimension 2^n. Unitary operators, which transform multiqubit states into other multiqubit states, serve as the quantum analogues of classical logic gates, such as AND, OR, NOT, and NAND. Diagrammatically, quantum algorithms are constructed by applying a sequence of gates (unitary operators) to wires (qubits) in a specific order, analogous to the circuit model in classical computing. It has been demonstrated that any unitary quantum operation can be decomposed into a combination of single-qubit gates and two-qubit gates [8].

Proc. of the 2025 Annual Modeling and Simulation Conference (ANNSIM'25)
May 26-29, 2025, Universidad Complutense de Madrid, Madrid, Spain
J.L. Risco-Martín, G. Rabadi, D. Cetinkaya, R. Cárdenas, S. Ferrero-Losada, and A.A. Abdelnabi, eds.

In comparison to classical circuits, quantum algorithms involve unique features of quantum mechanics, such as superposition and entanglement, which have no classical equivalent. And therefore, when evaluating a quantum algorithm or quantum circuit, one may consider how it could be simulated using only classical computation. It is clear that this task becomes challenging due to the exponential growth of the state space when the number of qubits increases. However, it has been proved a class of quantum circuits, involving Clifford gates [9], can be efficiently simulated classically [10]. A review of various approaches to the quest of simulating classically quantum computations is found in Ref. [11]. Another approach instead of fully simulate a quantum circuit involves the use of classical computation for pre- and/or post-processing steps [12].

An important aspect of quantum computer simulation is the classical model for representing a quantum circuit. In the literature, there exists circuit representations based on algebraic expressions [13, 14, 15]. In particular, complex Clifford Algebras have been proposed for representing qubits and quantum gates through the geometric product [13], which for any two elements u and v of the algebra has the following form

$$uv = u \cdot v + u \wedge v, \tag{2}$$

where the product $\cdot$ stands for scalar product, while the external product $\wedge$ is antisymmetric. As shown in Ref. [13], elements of the Witt basis, definable in finite-dimensional complex Clifford algebras, are used to define the state vectors necessary to algebraically represent all the qubits of a given quantum computer. The Witt basis is characterized by elements f_i and $f_i^\dagger$, which fulfill the following anticommutation relations (for details see Ref. [16]),

$$\left\{f_i\,, f_j\right\} = \left\{f_i^\dagger, f_j^\dagger\right\} = 0, \quad \left\{f_i\,, f_j^\dagger\right\} = \delta_{ij}. \tag{3}$$

Note that the product on the above equation is just the geometric product of the complex Clifford algebra. It is noteworthy that, the algebra elements f_i and $f_i^\dagger$ behave like the fermionic annihilation and creation operators in physics. In the simplest complex Clifford algebra, where only a pair of $f, f^\dagger$ exists, the state vector $|0\rangle$ and $|1\rangle$ may be defined as

$$|0\rangle := ff^\dagger, \quad |1\rangle := f^\dagger\,|0\rangle = f^\dagger,$$

among many other choices. A qubit is then just given by (1). The scalar product of different qubit states is defined within the algebra as $\langle a|b\rangle = \frac{1}{2}(ab)_0$, where the notation $(a)_0$ denotes the projection to the scalar (grad zero) part of the algebra element a. In Ref. [13], it is also shown that any unitary transformation acting on a qubit can be expressed in terms of these operators. For example, if one considers the quantum equivalent of the NOT gate, which is the Pauli matrix X, the quantum gate X is then represented as

$$X = f + f^\dagger.$$

Remarkably, within this framework [13], the geometric product is used not only to realize the product among unitary operators, but also to represent the application of a gate to a qubit state.

The generalization to multiqubit states is straightforward, namely, one defines the state $|00\cdots 0\rangle$ as

$$|00\cdots 0\rangle := |0\rangle_1 \otimes |0\rangle_2 \otimes \cdots \otimes |0\rangle_n = \prod_{i=1}^{n} f_i f_i^\dagger.$$

One sees that the tensor product is also realized through the geometric product. In general, for any $\lambda_i \in \{f_i, f_i^\dagger, f_i f_i^\dagger, f_i^\dagger f_i\}$ the tensor product corresponds to

$$\lambda_1 \otimes \cdots \otimes \lambda_n \to (-1)^p\,\lambda_1 \cdots \lambda_n, \tag{4}$$

where p is the sum of all cardinals of the sets $S_i = \{j < i : \lambda_j = f_j \text{ or } \lambda_j = f_j^\dagger f_j\}$ in the case if $\lambda_i = f_i$ or $\lambda_l = f_i^\dagger$. This intricate formula arises from the fact that the order of each operator in the tensor product of operators acting on its respective subspace is commutative, whereas this is not true for the geometric product given in (2). Nonetheless, this complex Clifford algebra approach provides an elegant and unifying framework to represent a quantum circuit as a whole, and numerical computations have been carried out [17, 18]. In fact, the anticommutation relations given in (3) are remarkably powerful, as they are sufficient for performing calculations without the explicit determination of matrix elements - the so-called oscillator expansion technique. This characteristic is precisely what makes annihilation and creation operators so fundamental in quantum mechanics [19] and other areas of science [20, 21]. An example, where the the oscillator expansion technique relation were applied for computational algebraic simplifications can be found in Ref. [22].

The goal of the present paper is to show that the algebra of hard-core bosons provides an alternative way for representing qubits and quantum gates in a general quantum circuit. In contrast to the complex Clifford algebra construction [13], the hard-core boson algebra allows the realization of the tensor product of operators to be naturally commutative, which simplifies the determination of the parity factor $(-1)^p$ in (4). Once the hard-core boson algebra is delineated, the process of simulation is then explored.

The next sections of this paper are organized as follow. In the next section, one construct the hard-core boson algebra in the context of qubits. The design of quantum circuits is then introduced. In the Section 3 the representation of quantum gates based on products and sums of hard-core boson operators is established. Some applications are presented at Section 4. Finally, conclusions are drawn in Section 5.

2 THE ALGEBRA OF HARD-CORE BOSONS

In this section the algebra of the hard-core bosons and its application to represent quantum circuits are introduced. Hard-core bosons appeared motivated in many problems of condense matter physics. A few examples of application can be seen in [23, 24]. The purpose of this section is to show that they appear as a natural description of the qubit system.

As mentioned in the introduction, qubits are defined as the tensor product of multiple replicas of a two-dimensional quantum system, namely $\mathbb{C}^2$. A single qubit is a linear combination of the orthonormal basis $\{|0\rangle, |1\rangle\}$. One assumes that these two states are eigenstates of some observable, with corresponding eigenvalues being non-degenerate (distinguishable). This is usually denoted as the computational basis. Since the set $\{|0\rangle, |1\rangle\}$ is a basis, any linear operator acting on vector states in $\mathbb{C}^2$ can be fully decomposed in terms of four operators, namely, $|0\rangle\langle 0|$, $|1\rangle\langle 1|$, $|0\rangle\langle 1|$ and $|1\rangle\langle 0|$. The first two operators from the just mentioned list are the usual basis projectors, which sum gives the identity, as,

$$|0\rangle\langle 0| + |1\rangle\langle 1| = \mathbb{1},$$

while the remaining two operators are the so-called, the annihilation, a, and the creation, $a^\dagger$, operators, respectively,

$$a := |0\rangle\langle 1|, \quad a^\dagger := |1\rangle\langle 0|. \tag{5}$$

Their names arise from the following properties:

$$a|0\rangle = 0, \quad a|1\rangle = |0\rangle, \quad a^\dagger|0\rangle = |1\rangle, \quad a^\dagger|1\rangle = 0.$$

The projectors $|0\rangle\langle 0|$ and $|1\rangle\langle 1|$ can be written as products of annihilation and creation operators, as

$$aa^\dagger = |0\rangle\langle 0|, \quad a^\dagger a = |1\rangle\langle 1|. \tag{6}$$

Table 1: Closure of the operator composition law for the elements in $\{1, 0, a, a^\dagger, aa^\dagger, a^\dagger a\}$ (monoid).

$\cdot$	0	1	a	$a^\dagger$	$aa^\dagger$	$a^\dagger a$
0	0	0	0	0	0	0
1	0	1	a	$a^\dagger$	$aa^\dagger$	$a^\dagger a$
a	0	a	0	$aa^\dagger$	0	a
$a^\dagger$	0	$a^\dagger$	$a^\dagger a$	0	$a^\dagger$	0
$aa^\dagger$	0	$aa^\dagger$	a	0	$aa^\dagger$	0
$a^\dagger a$	0	$a^\dagger a$	0	$a^\dagger$	0	$a^\dagger a$

The combination $N := a^\dagger a$ is usually called the number operator and together with $M := aa^\dagger$, they satisfies:

$$N + M = \mathbb{1}, \quad NM = MN = 0.$$

The matrix representations of the operators N, M are idempotent Hermitian matrices with trace 1.

Taking into account the equations given in (5) and (6), one concludes that for any linear operator O it can be entirely described in terms of annihilation and creation operators,

$$O = O_{00}\, aa^\dagger + O_{01}\, a + O_{10}\, a^\dagger + O_{11}\, a^\dagger a. \tag{7}$$

with $O_{ij} := \langle i | O | j \rangle$. Directly from equations given in (5) and (6), one writes a closed relation between the annihilation and creation operators as

$$\left\{a, a^\dagger\right\} = \mathbb{1}, \quad a^2 = \left(a^\dagger\right)^2 = 0. \tag{8}$$

These operator properties form a noncommutative algebra, being $\mathbb{1}$ the identity element. This algebra is usually called Grassmann algebra by physicists. One sees in Table 1, that any product of annihilation and creation operators always leads to one of the elements of the set $\{1, 0, a, a^\dagger, aa^\dagger, a^\dagger a\}$.

In a multiqubit system of dimension 2^n one represents the basis states $\{|b\rangle\}$, with $b = 0, \ldots, 2^n - 1$ as tensor product of the two state quantum system $|0\rangle$ and $|1\rangle$ as

$$|b\rangle \equiv |b_1 b_2 \cdots b_n\rangle = |b_1\rangle \otimes |b_2\rangle \cdots \otimes |b_n\rangle,$$

where the elements b_i are the bits in the binary basis of the integer b (the bit on the far right b_n is the least significant bit), and therefore $b_i \in \{0, 1\}$. In the context of multiqubit vector space, it is natural to extend the definition of the annihilation and creation operators by the following definitions

$$a_i := \mathbb{1} \otimes \cdots \otimes \mathbb{1} \otimes a \ \otimes \mathbb{1} \otimes \cdots \otimes \mathbb{1},$$
$$a_i^\dagger := \mathbb{1} \otimes \cdots \otimes \mathbb{1} \otimes a^\dagger \otimes \mathbb{1} \otimes \cdots \otimes \mathbb{1},$$

where in the above equations each annihilation or creation is set at the position i in the tensor product string, while the other position are set to the identity. One can now deduce the algebra for the extended annihilation, a_i, and creation, $a_i^\dagger$, operators. Setting $i \neq j$, one has

$$\left[a_i, a_j\right] = \left[a_i^\dagger, a_j^\dagger\right] = \left[a_i, a_j^\dagger\right] = 0. \tag{9}$$

These relations above are typical of bosonic commutation relations. It means, when one multiplies operators from different qubit index, they commute and therefore the commutativity of tensor product of operators is preserved. This is a neat difference when comparing with (4) given in Section 1. The relation of annihilation and creation operators within the same qubit index, i.e., $i = j$, is instead given by anticommutation relations,

$$\left\{a_i, a_i^\dagger\right\} = \mathbb{1}, \quad \left\{a_i, a_i\right\} = \left\{a_i^\dagger, a_i^\dagger\right\} = 0, \tag{10}$$

exhibiting now a fermion-like behavior. This mixed behavior, or this inconsistency of the qubits with the properties of either bosons or fermions, is what is called in the literature as hard-core bosons and their algebra is fully determined by (9) and (10). The hard-core bosons are similar to the parafermions if one restricts oneself to qubit systems [25, 26].

The full basis of the multiqubit space can now be systematically enumerated. In what follows, one denotes the vacuum state vector $|\mathbf{0}\rangle := |00\cdots 0\rangle$. The transformation of the annihilation and creation operators when acting on the computational basis vectors is straightforward (in expressions like $|b_1 \cdots x \cdots b_n\rangle$ it is assumed that x is at the position i):

$$\begin{aligned} a_i\,|b_1\cdots 0\cdots b_n\rangle &= 0, \quad a_i^\dagger\,|b_1\cdots 0\cdots b_n\rangle = |b_1\cdots 1\cdots b_n\rangle\,, \\ a_i^\dagger\,|b_1\cdots 1\cdots b_n\rangle &= 0, \quad a_i\,|b_1\cdots 1\cdots b_n\rangle = |b_1\cdots 0\cdots b_n\rangle\,. \end{aligned}$$

Therefore any state vector $|b_1 b_2 \cdots b_n\rangle$ is written as

$$|b_1 b_2 \cdots b_n\rangle = \left(a_1^\dagger\right)^{b_1} \left(a_2^\dagger\right)^{b_2} \cdots \left(a_n^\dagger\right)^{b_n} |\mathbf{0}\rangle\,. \tag{11}$$

Again, here b_i stands for the value of the bit at position i. One can easily verify that such state vectors are orthogonal and properly normalized, i.e.,

$$\left\langle b'_1 b'_2 \cdots b'_n \middle| b_1 b_2 \cdots b_n \right\rangle = \delta_{b_1 b'_1}\,\delta_{b_2 b'_2} \cdots \delta_{b_n b'_n}.$$

For computations, it is useful to rewrite the equation given in (11) without introducing powers, as follows

$$|b_1 b_2 \cdots b_n\rangle = \left[b_1\, a_1^\dagger + (1-b_1)\mathbb{1}\right] \left[b_2\, a_2^\dagger + (1-b_2)\mathbb{1}\right] \cdots \left[b_n\, a_n^\dagger + (1-b_n)\mathbb{1}\right] |\mathbf{0}\rangle\,. \tag{12}$$

This derivation demonstrates that a hard-core boson structure naturally emerges from a multiqubit space, with the advantageous property of preserving tensor product commutativity. Since there is no need to correct the tensor parity, as required in (4), this implementation appears more efficient for constructing a simulator. However, for a large number of qubits, the exponential growth of the multiqubit space presents significant challenges for simulation.

Since between different indices all operators commute, the product of an arbitrary number of operators can be performed by associating operators with same indices together, maintaining their relative position. Further simplification can be performed by using Table 1. As an example, one has

$$a_5\, a_2^\dagger a_1\, a_2\, a_3^\dagger\, a_2^\dagger \;=\; a_1\, a_2^\dagger a_2 a_2^\dagger\, a_3^\dagger\, a_5 \;=\; a_1\, a_2\, a_3^\dagger\, a_5.$$

In order to compare two different expressions, one needs an ordering, which for example, one takes first the qubit index and then the creation operator before the annihilation operator (same index)

$$a_5\, a_1\, a_2\, a_3^\dagger\, a_2^\dagger \;=\; a_1\, a_3^\dagger\, a_5 - a_1\, a_2^\dagger a_2\, a_3^\dagger\, a_5.$$

Before concluding this section, one shows the equivalence or the relation of hard-core boson algebra to two well established frameworks, namely, the complex Clifford algebra and the Pauli strings.

Complex Clifford Algebras

One addresses now the question whether it is possible to map the hard-core boson algebra to the fermionic-like formulation in the context of complex Clifford algebras [13]. In fact, there is a well known mapping between hard-core bosons and fermions, the so-called Jordan-Wigner transformation [27], in which the annihilation and creation operators are transformed as

$$a_i \to f_i = \mathscr{J}_i a_i \quad \text{and} \quad a_i^\dagger \to f_i^\dagger = \mathscr{J}_i^\dagger a_i^\dagger ,$$

where $\mathscr{J}_i$ is an unitary operator defined by

$$\mathscr{J}_i := \prod_{k=1}^{i-1} e^{i\pi a_k^\dagger a_k}.$$

From the definition, one deduces the following relation

$$\mathscr{J}_i^\dagger = \mathscr{J}_i = \prod_{k=1}^{i-1} (-Z_k) ,$$

and if one takes into account that Z_i anticommutes with a_i and $a_i^\dagger$, it can then demonstrated that the resulted operators f_i and $f_i^\dagger$ fulfill the fermionic algebra given in (3), i.e.,

$$\left\{f_i\, , f_j^\dagger\right\} = \delta_{ij}\mathbb{1}, \quad \left\{f_i\, , f_j\right\} = \left\{f_i^\dagger , f_j^\dagger\right\} = 0.$$

The Jordan-Wigner transformations are know to be an elegant and systematic way to include fermionic properties into quantum computations [15].

Pauli Strings

One establishes first the map between the annihilation and creation operators and the X- and Y-Pauli matrices.

$$X_i = a_i + a_i^\dagger, \quad Y_i = i\,(a_i^\dagger - a_i), \tag{13}$$

with the converse being given by

$$a_i = \frac{X_i + iY_i}{2}, \quad a_i^\dagger = \frac{X_i - iY_i}{2}.$$

The Z-Pauli matrices are given by $Z_i := iX_iY_i$ and one has

$$Z_i = a_i^\dagger a_i - a_i a_i^\dagger = M_i - N_i.$$

There is a straightforward relation between Z_i and the number operator N_i and the operator M_i, as

$$N_i = \frac{\mathbb{1} - Z_i}{2}, \quad M_i = \frac{\mathbb{1} + Z_i}{2}.$$

As an example, for any 2-dimensional linear operator, O, the expression given in (7) can be rewritten as linear combination of Pauli matrices X, Y, Z, and the identity matrix $\mathbb{1}$, as

$$O = \frac{1}{2}\left[\,(O_{00} + O_{11})\,\mathbb{1} + (O_{10} + O_{01})\,X + i\,(O_{10} - O_{01})\,Y + (O_{00} - O_{11})\,Z\right].$$

Since the products of annihilation and creation operators are nothing more than a tensor product, it is not surprising that the definitions given in (13) for $i \neq j$ lead to

$$[X_i, X_j] = [Y_i, Y_j] = [Z_i, Z_j] = [X_i, Y_j] = [X_i, Z_j] = [X_i, Z_j] = 0,$$

which leads to the formalism of Pauli strings used extensively in quantum computation [28]. Pauli strings allows for a structured simulation process using classical computations [14].

3 QUANTUM CIRCUITS

Table 2: The 1-qubit quantum gates written in terms of creation and annihilation operators. Note that in the table qubit indices were dropped for sake of simplification.

Quantum Circuits	Operator Sequences
X	$a^\dagger + a$
Y	$i(a^\dagger - a)$
Z	$aa^\dagger - a^\dagger a$
H	$\frac{1}{\sqrt{2}}\left(aa^\dagger - a^\dagger a + a^\dagger + a\right)$
S	$aa^\dagger + ia^\dagger a$
T	$aa^\dagger + e^{i\frac{\pi}{4}} a^\dagger a$
$R_x(\theta)$	$\cos\left(\frac{\theta}{2}\right)\left(aa^\dagger + a^\dagger a\right) + i\sin\left(\frac{\theta}{2}\right)\left(a + a^\dagger\right)$
$R_y(\theta)$	$\cos\left(\frac{\theta}{2}\right)\left(aa^\dagger + a^\dagger a\right) + \sin\left(\frac{\theta}{2}\right)\left(a - a^\dagger\right)$
$R_z(\theta)$	$e^{i\frac{\theta}{2}} aa^\dagger + e^{-i\frac{\theta}{2}} a^\dagger a$

From the previous section, one can derive from (7) any 1-qubit gate, since it is an unitary operator. In Table 2 an example of 1-qubit gates are listed. In the case of multiqubit gates, an extension of (7) can easily be done. In the multiqubit space 2^n-dimensional, an unitary operator acting on the space can be interpreted as

$$U = \sum_{b,b'=0}^{2^n-1} U_{bb'} \left|b\right\rangle \left\langle b'\right|,$$

each operator $|b\rangle\langle b'|$ can then be written as

$$\left|b\right\rangle \left\langle b'\right| = \left|b_1 \cdots b_n\right\rangle \left\langle b'_1 \cdots b'_n\right| = \left|b_1\right\rangle \left\langle b'_1\right| \otimes \cdots \otimes \left|b_n\right\rangle \left\langle b'_n\right|.$$

Then, for each $|b_1\rangle\langle b'_1|$ one has from (5) and (6) the following identification:

$$|0\rangle\langle 0| = a_i a_i^\dagger, \quad |0\rangle\langle 1| = a_i, \quad |1\rangle\langle 0| = a_i^\dagger, \quad |1\rangle\langle 1| = a_i^\dagger a_i.$$

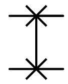
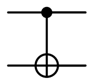
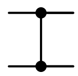
As illustration, if ones take a general control gate, $\Lambda^1(U) = |0\rangle\langle 0| \otimes \mathbb{1} + |1\rangle\langle 1| \otimes U$, between the first and second qubit, one gets written in terms of annihilation and creation operators as

$$\Lambda^1(U) = a_1 a_1^\dagger + a_1^\dagger a_1 \Big(U_{11} a_2 a_2^\dagger + U_{12} a_2 + U_{21} a_2^\dagger + U_{22} a_2^\dagger a_2\Big),$$

where U is an unitary operator acting on $\mathbb{C}^2$. The Table 3 shows examples of two-qubit gates written with only terms involving annihilation and creation operators.

In what follows, one demonstrates the powerful technique of using annihilation and creation operators to simulate probabilities for a given circuit. As in standard quantum machines, one begins with the state of the computation by assuming the system is initially set to the vacuum state $|\mathbf{0}\rangle$. This is not a restriction,

Table 3: The 2-qubit quantum gates written with creation and annihilation operators.

Quantum Circuits	**Operator Sequences**
	$a_1 a_1^\dagger a_2 a_2^\dagger + a_1^\dagger a_1 a_2^\dagger a_2 + a_1^\dagger a_2 + a_1 a_2^\dagger$
	$a_1 a_1^\dagger + a_1^\dagger a_1 \left(a_2^\dagger + a_2\right)$
	$a_1 a_1^\dagger + a_1^\dagger a_1 \left(a_2 a_2^\dagger + a_2^\dagger a_2\right)$

since any initial state can be prepared and included at the beginning of the gate sequence. To make sense of the described hard-core boson formalism for simulating a given quantum circuit, some assumptions must be made. Thus, for a given sequence of gates $G := G_p G_{p-1} \cdots G_1$, one assumes these gates to operate sequentially on the initial state $|\mathbf{0}\rangle$, and at the end of the operation, the initial state evolves into a final state $|f\rangle$. It is also assumed that during the application of each gate G_i to the resulting intermediate state, no measurements are performed. Only after the full sequence of gates is applied does the measurement process occur, marking the end of the computation. Finally, for sake of simplicity, it is assumed that the quantum simulation operates under ideal conditions, without considering any source of errors.

Within this algebraic framework, the effective resulting gate G is calculated by using the operations allowed by the hard-core boson algebra. The strength of this method is to calculate G without requiring the determination of any matrix elements for the operators involved, relying solely on the rules given in (9) and (10). The number of matrix elements grows exponentially as 2^{2n} for n qubits and each gate, G_i, action, on a qubit subset, would need to be represented in separated matrix blocks. However, within this method, no adjustments of this kind are required.

It is quite natural in quantum simulations to determine the final state vector just before the final measurement. In a real quantum computing device, the final measurement is the only means to extract information about the final state. The circuit must be applied repeatedly to obtain the probability distribution of the final state. Theoretically, the probability that the final state vector is in the state $|b_1 b_2 \cdots b_n\rangle$ is given by

$$P(b|G) := P(b_1 b_2 \cdots b_n | G_p G_{p-1} \cdots G_1) = |\langle b_1 b_2 \cdots b_n | \, G \, |\mathbf{0}\rangle|^2 .$$

Using the expression given in (12) in the bracket of the above equation, the probability $P(b, G)$ becomes the expectation value of an algebraic expression (defined in the hard-core boson algebra) with respect to $|\mathbf{0}\rangle$, as

$$P(b|G) = |\langle \mathbf{0}| \, [b_1 \, a_1 + (1 - b_1) \, \mathbb{1}] \cdots [b_n \, a_n + (1 - b_n) \, \mathbb{1}] \, G \, |\mathbf{0}\rangle|^2 . \tag{14}$$

The algebraic rules given in (9) and (10) allow to move the creation operators to the left of the expression, and the annihilation operators to the right. Any term with an annihilation operator vanishes when applied to state $|\mathbf{0}\rangle$ and any creation operator also vanishes when applied to the dual state $\langle \mathbf{0}|$. The only terms that persist are terms proportional to the identity, due to the Kronecker tensor in the anticommutator relations given in (10). This demonstrates the potential of this algebraic technique for simulating quantum circuits.

It is worth to point out that one can simplify the scheme above with the aim to extract the probabilities without systematically preparing the dual state $\langle b_1 b_2 \cdots b_n|$. Before the set of gates G acts on the state $|\mathbf{0}\rangle$, one should order the operators in order to move all annihilation to the right. As in the previous case, the $|\mathbf{0}\rangle$ will vanish all terms containing annihilation operators, resulting in the following equation

$$|f\rangle := G_p G_{p-1} \cdots G_1 \, |\mathbf{0}\rangle = \left(\sum_{b=0}^{2^n - 1} \zeta_b \, (a_1^\dagger)^{b_1} \, (a_2^\dagger)^{b_2} \cdots (a_n^\dagger)^{b_n} \right) |\mathbf{0}\rangle \, , \tag{15}$$

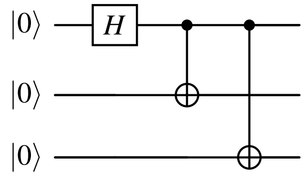


Figure 1: The Greenberger–Horne–Zeilinger state quantum circuit.

where ζ_b are just complex coefficients. For each b and its binary decomposition $b = (b_1, b_2 \dots b_n)$, one can read the probability directly as

$$P(b|G) = |\zeta_b|^2.$$

The operator inside of the parenthesis of the right-handed is no longer unitary, but it has the necessary information to reconstruct the probabilities. The resulting state is then given by

$$|f\rangle = \sum_{b=0}^{2^n-1} \zeta_b \, |b_1 b_2 \dots b_n\rangle .$$

4 APPLICATIONS

In what follows three examples are given to show the potential of the circuit representation and simulation in the context of hard-core bosons: preparation of the Greenberger–Horne–Zeilinger state through a quantum circuit, simulation a Grover algorithm and the question how to compare two given quantum circuits. Finally, computational developments for implementing this hard-core boson framework is reported.

The Greenberger–Horne–Zeilinger state

In this example, the Greenberger–Horne–Zeilinger (GHZ) state for 3 qubits is considered. In quantum information, the GHZ state is a certain type of entangled quantum state that involves at least three [29] or more qubits [30]. The 3-qubit GHZ state has been observed [31]. They play a crucial role in fundamental tests of quantum mechanics versus local realism and in many quantum information and quantum computation schemes [32]. In Figure 1, one shows a quantum circuit prepares the GHZ state.

Using the relation given in (15) for the circuit given in Figure 1, on gets

$$G_{\text{GHZ}} \, |\mathbf{0}\rangle = \left(\frac{1}{\sqrt{2}} \mathbb{1} + \frac{1}{\sqrt{2}} a_1^\dagger a_2^\dagger a_3^\dagger \right) |\mathbf{0}\rangle .$$

As expected from Section 3, the right-handed side of the above equation is no longer a unitary operator acting on the state $|\mathbf{0}\rangle$, but leads to the same final vector state,

$$|\text{GHZ}\rangle = \frac{|000\rangle + |111\rangle}{\sqrt{2}}.$$

One reads from the GHZ state all nonvanishing probabilities, which are $P(000|G_{\text{GHZ}}) = P(111|G_{\text{GHZ}}) = \frac{1}{2}$.

Grover example

In this example, the hard-core boson formalism is applied to simulate a Grover algorithm [33]. The goal of the Grover algorithm is to find the solution state $|\omega\rangle \in \mathbb{C}_2^{\otimes n}$ which is marked by the oracle U_ω via a phase

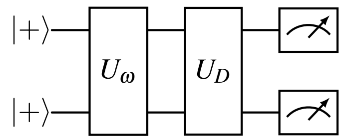


Figure 2: Quantum circuit for the Grover algorithm to find the solution $|\omega\rangle = |10\rangle$ marked by the oracle U_ω, followed by the action of the diffusor U_D (the reflection across the initial state). Note that since $n = 2$ qubits are used, the Grover iteration $U_D U_\omega$ is applied only once.

flip, i.e., $U_\omega |\omega\rangle = -|\omega\rangle$. To illustrate the hard core boson formalism, the Grover algorithm is simulated on $n = 2$ qubits for $|\omega\rangle = |10\rangle$. The circuit is depicted in Figure 2 and the involved operations are

$$U_\omega = \mathbb{1} - 2|\omega\rangle\langle\omega| = 1 - 2a_1^\dagger a_1 a_2 a_2^\dagger$$
$$U_D = 2|+\rangle\langle+| - \mathbb{1} = \frac{1}{4}(1 + a_1 + a_1^\dagger)(1 + a_2 + a_2^\dagger) - 1.$$

The action of the circuit is

$$C := U_D U_\omega H^{\otimes 2} = a_1 a_2^\dagger a_2 + a_1^\dagger - a_1^\dagger a_2^\dagger a_2 + a_1^\dagger a_1 a_2 + a_1^\dagger a_1 a_2^\dagger - a_2,$$

where the Hadamard state $|+\rangle := (|0\rangle + |1\rangle)/\sqrt{2}$ preparation was also included. From the procedure to deduce the final state in (15) in the Section 3, one gets $C|\mathbf{0}\rangle = a_1^\dagger |\mathbf{0}\rangle$, which lead to the outcome probability

$$P(2|C) = |\langle\mathbf{0}|a_1 C|0\rangle|^2 = |\langle\mathbf{0}|a_1 a_1^\dagger|\mathbf{0}\rangle|^2 = 1.$$

As expected, the solution $|\omega\rangle = |01\rangle$ is correct with a prediction for the measurement of 100% of certainty.

Circuit comparison

The methods described in this section can also be used to compare two distinct quantum circuits. Assuming all gates involved in the circuit are unitary, the two circuits, with respective final unitaries G and G', are equivalent if and only if

$$\left|\langle\mathbf{0}|\, G^\dagger\, G'\, |\mathbf{0}\rangle\right|^2 = 1.$$

Similar to the procedure made on the equation given in (14), the ordering of creation operators to the left and annihilation operators to the right ensures that all terms vanish, except those proportional to the identity. This simplifies the evaluation of the bracket, making it calculable within the hard-core boson formalism.

Computational implementation of hard-core bosons

In order to test the viability of the oscillator expansion techniques based on hard-core bosons for representing larger quantum circuits, a library written in C++ is being developed [34]. For performing the test, two benchmark circuits from the Munich Quantum Toolkit Benchmark Library (MQT Bench) [35] were chosen, namely, the quantum Fourier transformation algorithm (QFT) and the Grover's algorithm (v-chain). For both algorithms, the target-dependent native IBM gates *rz*, *sx*, *x* and *cx* were selected from the MQT Bench interface and delivered as OpenQASM 2.0 files [36]. The QFT circuit was generated for 5 qubits (composed of a total 71 gates), while the Grover's circuit was generated for 9 qubits (composed of a total of 3187 gates).

For this test, the statevector simulator of the Quantum Information Software Kit (Qiskit) [37] of version 1.4.2 was used as the reference for comparison of the performance of the two algorithms. Both Qiskit and

Quipo tools were run in the chip Apple M1 (2020) with 16*GB* of RAM and under macOS Sequoia 15.3.2 (`C++` compiler: Apple clang version 16.0.0). As a result, the QFT algorithm performed 0.0693*ms* on Quipo and 2.35*ms* on Qiskit, while the Grover's algorithm performed 55.7*ms* on Quipo and 98.6*ms* on Qiskit. The probabilities of the final state for each algorithm are identical for both Quipo and Qiskit. It was observed that the advantage of the Quipo over Qiskit depends strongly on the circuit topology. A more comprehensive benchmark analysis comparing Quipo with Qiskit and/or other statevector simulators is definitely needed.

5 CONCLUSION

Quantum computing has emerged as a multidisciplinary field combining computer science, physics, and mathematics, offering the potential to simulate quantum systems without the exponential overhead of classical simulations. Central to standard quantum computing is the concept of quantum circuits, composed of qubits and quantum gates, providing a common language among researchers from various fields to understand and develop quantum algorithms. Quantum algorithms exploit quantum mechanical features like superposition and entanglement, which have no classical analogs, making quantum system simulations particularly challenging. Nonetheless, certain classes of quantum circuits, such as Clifford circuits, have been demonstrated to be efficiently simulable on classical computers.

An important aspect of quantum simulation is the representation of the elements of a quantum computer. Algebraic constructions based on complex Clifford algebra [13] provide a potential unification of all components of a quantum circuit (such as qubits, gates, and the tensor product) into two concepts: algebraic elements and the geometric product. However, because the tensor product is commutative, a parity correction is necessary due to the fact that the geometric product is inherently non-Abelian.

In this paper, the algebra of hard-core bosons was explored in order to improve the tensor product representation originally formulated by the complex Clifford algebra [13]. This new approach realizes the tensor product without the need for parity corrections, since the new product is commutative for elements with different indices. It was also observed that multiqubit systems naturally aligns with the algebra of hard-core bosons, making this novel algebraic framework well-suited for quantum simulations. The Jordan-Wigner transformation was evoked in order to demonstrate the equivalence of two frameworks, the former based on complex Clifford algebras and the latter based on the algebra of the hard-core-bosons.

It was also shown in this paper how to simulate a quantum computing process and calculate the probabilities of the final state for a given algorithm. The methods presented in the paper strongly demonstrate the power of the algebraic approach, which relies on the presence of annihilation and creation operators, without the need for explicit determination of matrix elements, a process that grows exponentially with the number of qubits. Since the tensor product is realized more simply within this framework, it is expected to lead to more efficient computational tools (like the Quipo library [34]), thus enhancing the simulation of quantum systems.

ACKNOWLEDGMENTS

The author is thankful to Alexander Kegeles for contributing to the first concepts of this work; to Thomas Keitzl for the encouragement and energy; to Peter K. Schuhmacher, Thomas Stehle and Johannes Renkl for enthusiastic discussions and ideas. Deep gratitude is also expressed to Michael Epping for his inestimable help for proofreading and verifying the paper.

This project was made possible by the DLR Quantum Computing Initiative and the Federal Ministry for Economic Affairs and Climate Action; qci.dlr.de/projects/ALQU.

REFERENCES

[1] R. P. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys.*, vol. 21, pp. 467–488, 1982.

[2] P. Benioff, "The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines," *Journal of Statistical Physics*, vol. 22, no. 5, pp. 563–591, 1980.

[3] Y. I. Manin, *The computable and the non-computable (Vychislimoe i nevychislimoe)*. Kibernetika. Moskva: Sovetskoe RadioSovetskoe Radio, 1980, r. 0.45.

[4] D. E. Deutsch, "Quantum computational networks," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 425, no. 1868, pp. 73–90, Sep. 1989.

[5] F. Arute, K. Arya, R. Babbush, and et al., "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, pp. 505–510, 2019.

[6] A. Basermann, M. Epping, B. Fauseweh, M. Felderer, E. Lobe, M. Röhrig-Zöllner, G. Schmiedinghoff, P. K. Schuhmacher, Y. Setyawati, and A. Weinert, "Quantum Software Ecosystem Design," 5 2024.

[7] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug 2018.

[8] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, "Elementary gates for quantum computation," *Physical Review A*, vol. 52, no. 5, pp. 3457–3467, 1995.

[9] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Phys. Rev. A*, vol. 70, no. 5, p. 052328, 2004.

[10] D. Gottesman, "The heisenberg representation of quantum computers," in *Proceedings of the 22nd International Colloquium on Group Theoretical Methods in Physics*, 1998. [Online]. Available: https://arxiv.org/abs/quant-ph/9807006

[11] A. Cicero, M. A. Maleki, M. W. Azhar, A. F. Kockum, and P. Trancoso, "Simulation of quantum computers: Review and acceleration opportunities," 2024. [Online]. Available: https://arxiv.org/abs/2410.12660

[12] M. Epping, "Hybrid simplification rules for boundaries of quantum circuits," 6 2022.

[13] J. Hrdina, A. Navrat, and P. Vasik, "Quantum computing based on complex Clifford algebras," *Quant. Inf. Proc.*, vol. 21, no. 9, p. 310, 2022.

[14] G. Biswas, "On Classical Simulation of Quantum Circuits Composed of Clifford Gates," *Quanta*, vol. 13, no. 1, pp. 20–27, 2024.

[15] G. Veyrac and Z. Toffano, "Geometric Algebra Jordan–Wigner Transformation for Quantum Simulation," *Entropy*, vol. 26, no. 5, p. 410, 2024.

[16] F. Brackx, H. De Schepper, and F. Sommen, "The Hermitian Clifford Analysis Toolbox," *Adv. Appl. Clifford Algebras*, vol. 18, no. 3-4, pp. 451–487, 2008.

[17] J. Hrdina, D. Hildenbrand, A. Návrat, C. Steinmetz, R. Alves, C. C. Lavor, P. Vašík, and I. Eryganov, "Quantum Register Algebra: the mathematical language for quantum computing," *Quant. Inf. Proc.*, vol. 22, no. 9, p. 328, 2023.

[18] J. Hrdina, D. Hildenbrand, A. Návrat, C. Steinmetz, R. Alves, C. Lavor, P. Vašík, and I. Eryganov, "Quantum register algebra: the basic concepts," *Lect. Notes Comput. Sci.*, vol. 13771, p. 112, 2023.

[19] P. A. M. Dirac, "Quantum theory of emission and absorption of radiation," *Proc. Roy. Soc. Lond. A*, vol. 114, p. 243, 1927.

[20] R. N. Mohapatra and B. Sakita, "SO(2n) Grand Unification in an SU(N) Basis," *Phys. Rev. D*, vol. 21, p. 1062, 1980.

[21] F. Wilczek and A. Zee, "Families from Spinors," *Phys. Rev. D*, vol. 25, p. 553, 1982.

[22] N. Cardoso, D. Emmanuel-Costa, N. Gonçalves, and C. Simoes, "SOSpin, a C++ library for Yukawa decomposition in SO(2N) models," *Comput. Phys. Commun.*, vol. 203, pp. 178–200, 2016.

[23] T. Matsubara and H. Matsuda, "A lattice model of liquid helium, i," *Progress of Theoretical Physics*, vol. 16, no. 5, pp. 569–582, 1956.

[24] P. W. Anderson, "Absence of diffusion in certain random lattices," *Physical Review*, vol. 109, no. 5, pp. 1492–1505, 1958.

[25] D. Bonatsos, C. Daskaloyannis, and K. Kanakoglou, "Parafermionic and generalized parafermionic algebras," in *International Workshop on Quantum Groups, Deformations and Contractions*, 9 1999.

[26] L. A. Wu and D. A. Lidar, "Qubits as parafermions," *J. Math. Phys.*, vol. 43, no. 9, pp. 4506–4525, 2002.

[27] P. Jordan and E. P. Wigner, "About the Pauli exclusion principle," *Z. Phys.*, vol. 47, pp. 631–651, 1928.

[28] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[29] N. D. Mermin, "Quantum mysteries revisited," *American Journal of Physics*, vol. 58, no. 8, pp. 731–734, Aug. 1990.

[30] D. M. Greenberger, M. A. Horne, and A. Zeilinger, "Going Beyond Bell's Theorem," *Fundam. Theor. Phys.*, vol. 37, pp. 69–72, 1989.

[31] D. Bouwmeester, J.-W. Pan, M. Daniell, H. Weinfurter, and A. Zeilinger, "Observation of three photon Greenberger-Horne-Zeilinger entanglement," *Phys. Rev. Lett.*, vol. 82, pp. 1345–1349, 1999.

[32] M. Epping, H. Kampermann, C. macchiavello, and D. Bruß, "Multi-partite entanglement can speed up quantum key distribution in networks," *New J. Phys.*, vol. 19, no. 9, p. 093012, 2017.

[33] M. Epping, private communication, 2025.

[34] D. Emmanuel-Costa, T. Keitzl, and M. Röhrig-Zöllner, "Quipo, a c++ library for implementing hard-core bosons algebra," 2025, under development, Institute of Software Technology, German Aerospace Center (DLR), Cologne, Germany.

[35] N. Quetschlich, L. Burgholzer, and R. Wille, "MQT Bench: Benchmarking Software and Design Automation Tools for Quantum Computing," *Quantum*, 2023, MQT Bench is available at https://www.cda.cit.tum.de/mqtbench/.

[36] A. W. Cross, L. S. Bishop, J. A. Smolin, and J. M. Gambetta, "Open Quantum Assembly Language," 7 2017.

[37] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, "Quantum computing with Qiskit," 2024.

AUTHOR BIOGRAPHIES

DAVID EMMANUEL-COSTA is a researcher at the High Performing Computing (HPC) of the Institute of Software Technology (SC) of the German Aerospace (DLR), Cologne, Germany. He obtained a Ph.D. in Theoretical Particle Physics in University of Lisbon (UL) in Portugal, he worked as Postdoc at Deutsches Elektronen-Synchrotron (DESY), Hamburg, Germany and at UL, Portugal. His research interests include optimization and simulation in quantum computing; topics on particle physics; collective neutrino oscillations on quantum computer. His email address is david.dacosta@dlr.de.