Simulating thermal qubits through thermofield dynamics: an undergraduate approach using quantum computing

G.X.A. Petronilo[1], M.R. Araújo[2], Clebson Cruz[*2]

[1]Universidade de Brasília, Centro Internacional de Física, Instituto de Física, Brasília, DF, Brasil.
[2]Universidade Federal do Oeste da Bahia, Centro de Ciências Exatas e das Tecnologias, Grupo de Informação Quântica e Física Estatística, Barreiras, BA, Brasil.

Quantum computing has attracted the attention of the scientific community in the past few decades. The development of quantum computers promises one path toward safer and faster ways to treat, extract, and transfer information. However, despite the significant advantages of quantum computing, the development of quantum devices operating at room temperature has been compromised by the thermal decoherence process. In addition, in most undergraduate and graduate quantum mechanics courses, the study of thermofield dynamics is usually neglected. In this scenario, this work presents a didactic approach to simulate thermal qubit systems through Thermofield Dynamics (TFD), applied in a quantum computing setup. The results show that the Bloch sphere representation for a qubit can be written in terms of the Bogoliubov transformation, which allows a practical construction for the thermal qubits in a quantum computing setup. Therefore, this work introduces thermofield dynamics through quantum computing to teachers and curious students interested in teaching and learning this important field of studying the temperature impacts on quantum protocols using the TFD technique.
Keywords: Quantum Computing, Thermofield Dynamics, IBM Quantum Experience.

1. Introduction

In the past few years, we have seen the emergence of several technologies based on quantum properties of advanced materials [1–10]. However, the advantages of these quantum technologies are compromised since it is impossible to decouple the system from its surroundings and the decoherence phenomena are inevitable [11]. Quantum features such as entanglement [12] and coherence [13], for instance, are extremely sensitive to thermal coupling, compromising their usability at room temperature and, consequently, the development of quantum technologies under room conditions [9, 11, 12].

As a consequence, the study of thermal effects on the dynamics of quantum systems has attracted the attention of the scientific community [14–22]. In this scenario, the Thermofield Dynamics (TFD) appears as a useful approach since it allows the treatment of temporal and thermal contributions equally [14, 16–19, 23–26]. Therefore, TFD is a natural approach to studying quantum states at finite temperatures. In contrast to other finite-temperature quantum field theories, which employ imaginary-time based on path integrals, the TFD formalism differs by doubling the degrees of freedom in a Hilbert space and applying a temperature-dependent Bogoliubov transformation [27]. The application of higher-dimensional Hilbert spaces has been

used for simplifying quantum logic in the construction of quantum circuits [28]. On the other hand, the TFD approach gives an advantage in studying gate-based quantum computers because it is a real-time operator-based approach, using Bogoliubov transformations of quantum field theory at finite temperature [23].

In this context, this theory has shown great potential for studying the thermal properties of qubit systems regarding the development of quantum protocols at finite temperatures [14, 17, 23–26, 29]. In addition, applying higher-dimensional Hilbert spaces have been used to simplify quantum logic in the construction of quantum circuits [28]. Implementing qubits and logical gates at finite temperature using TFD formalism has long been a question of great interest in the past few years [14, 17, 23, 24, 26, 29, 30]. As a result, the realization of simulation of thermal qubits using variational methods was developed [17, 23, 24, 30], and recently it was demonstrated that the verification of TFD doubled state on a trapped-ion quantum computer [25]. Furthermore, since quantum computers can model other quantum mechanical systems, the preparation of thermal equilibrium states in quantum computers can be used to study temperature-dependent quantum phenomena [14, 19].

In this context, this paper shows a teaching approach for simulating thermal qubit systems using Thermofield Dynamics (TFD) in a quantum computing environment. The primary objective is to present a proof-of-concept for implementing thermal qubits in a quantum

---

*Correspondence email address: clebson.cruz@ufob.edu.br

computing configuration using the TFD theory, in a manner that is accessible to undergraduate students with an interest in this important and expansive field of quantum physics. We assume that the readers have knowledge of the well-known quantum mechanics and quantum computing fundamentals [31] Exploiting that the TFD formalism uses dual states to express temperature dependence [23], we show how to build a Bloch sphere representation for a two-level quantum system (qubit) in terms of the Bogoliubov transformations. The thermofield-double qubits were simulated using the quantum information software development kit – *IBM Qiskit*, available on the *IBM Quantum Experience* platform [32–35], available free of charge and easily accessible to students.

Using the TFD algebraic approach, we show the implementation of the famous quantum teleport algorithm at finite temperatures. The results show that using Bloch sphere representation for a qubit, written in terms of the Bogoliubov transformation, allows the construction of thermal qubits in a quantum computing setup, making it more intuitive and approachable for quantum algorithm developers. Consequently, the algebraic method of TFD theory makes modeling quantum protocols at finite temperatures easier for quantum algorithm developers, lowering the number of quantum gates required. This decrease in gates simplifies quantum logic opening a large avenue for the teaching and learning of thermal effects on quantum protocols. Therefore, this work provides an enticing introduction to thermofield dynamics via quantum computing, aimed at undergraduate teachers, and inquisitive undergraduate and graduate students, interested in teaching and learning this significant and extensive field of researching the thermal effects on quantum protocols. The codes used in this article are accompanied by detailed protocols throughout the text[1], enabling readers to replicate and apply them to their own quantum computing initiatives, even if they have limited experience in scientific computing.

2. Thermofield Dynamics

Thermofield Dynamics (TFD) is an approach to quantum field theory at finite temperature [19, 27, 37, 38]. Different from other finite-temperature quantum field theories, based on fictitious imaginary times [37, 38], the TFD formalism emerges as a natural way of describing the thermal phenomena in a real-time operator-based approach through the doubling of the Hilbert space and the applying of the Bogoliubov transformations [27, 37, 38].

In this scenario, the tilde $(\sim)$ conjugate rules give the doubling of Hilbert space, where the thermal space can be defined as $\mathcal{S}_T = \mathcal{S} \otimes \widetilde{\mathcal{S}}$, with S being the standard

---

Hilbert space and $\widetilde{S}$ being the tilde (dual) space [27, 38]. This doubling is defined by the mapping $(\sim) : \mathcal{S} \to \widetilde{\mathcal{S}}$, which connects each operator in S to each operators in $\widetilde{S}$, such as

$$\mathcal{A} = a \otimes 1, \qquad \widetilde{\mathcal{A}} = 1 \otimes a . \tag{1}$$

Thus, the relationship between the tilde $\widetilde{A}_i$ and non-tilde A_i operators can be defined as:

$$(\mathcal{A}_i \mathcal{A}_j)^\sim = \widetilde{\mathcal{A}_i} \widetilde{\mathcal{A}_j}, \quad (c\mathcal{A}_i + \mathcal{A}_j)^\sim = c^* \widetilde{\mathcal{A}_i} + \widetilde{\mathcal{A}_j},$$
$$(\mathcal{A}_i^\dagger)^\sim = \widetilde{\mathcal{A}_i}^\dagger, \qquad (\widetilde{\mathcal{A}_i})^\sim = -\xi \mathcal{A}_i, \tag{2}$$

with $\xi = -1$ for bosons and $\xi = +1$ for fermions [27, 37, 38].

On the other hand, a Bogoliubov transformation, $U(\alpha)$, is used to introduce thermal effects by inducing a rotation between the tilde and non-tilde variables, and it is defined as

$$U(\alpha) = \begin{pmatrix} u(\alpha) & -v(\alpha) \\ \xi v(\alpha) & u(\alpha) \end{pmatrix}, \tag{3}$$

where $u^2(\alpha) + \xi v^2(\alpha) = 1$. The α parameter is defined as the compactification parameter given by $\alpha = (\alpha_0, \alpha_1, \cdots, \alpha_{D-1})$. The temperature effect can be algebrically described by choosing $\alpha_0 \equiv \beta$ and $\alpha_1, \cdots \alpha_{D-1} = 0$, where $\beta \propto \frac{1}{T}$ with T being the temperature.

2.1. Boson representation for the SU(2) algebra

In order to introduce the concept of a thermal qubit, one can use an alternative algebraic approach to studying two-level systems as spin-$1/2$, for instance, through a boson representation of SU(2) algebra [27]. In this context, the ladder operators can be defined as

$$S^\pm = (\sigma^x + i\sigma^y), \qquad \text{and} \qquad S^0 = \sigma^z . \tag{4}$$

Consequently, for each spin variable, we get

$$[S^0, S^\pm] = \pm S^\pm, \qquad [S^+, S^-] = 2S^0 . \tag{5}$$

Then one can define

$$S^+ = a_1^\dagger a_2, \quad S^- = a_2^\dagger a_1, \quad \text{and} \quad S_0 = \frac{1}{2}(a_1^\dagger a_1 - a_2^\dagger a_2) , \tag{6}$$

where a_1 and a_2 satisfying the double boson algebra

$$[a_i, a_j^\dagger] = \delta_{i,j} \tag{7}$$

with $i, j = 1, 2$, and all other commutations being zero.

Thus, the connection to the original SU(2) algebra can be recovered assuming $n_1 = s + m$, $n_2 = s - m$, where $n_i = a_i^\dagger a_i$, with s and m are related to the usual results

$$\sigma^2 |s, m\rangle = s(s+1)|s, m\rangle,$$
$$\sigma_z |s, m\rangle = m|s, m\rangle \tag{8}$$

The rules for applying $S^{\pm}$ operators to states are as follows:

$$S^{-}|0,1\rangle = 0, \qquad S^{+}|1,0\rangle = 0,$$
$$S^{-}|1,0\rangle = |0,1\rangle, \qquad S^{+}|0,1\rangle = |1,0\rangle, \qquad (9)$$

It is worth noting that, $S^{-}|0,1\rangle = 0$, i.e. $|0,1\rangle$ is the vacuum state for S^{-}. Due to the fact that this represents a two-level system, one can represent a ground state and an excited state from an energetic point of view. In this context, the corresponding basis of this Hilbert space can be redefined as

$$|0,1\rangle = |0\rangle, \qquad |1,0\rangle = |1\rangle,$$

where $|0\rangle$ is the ground state, and $|1\rangle$ is the excited one. Thus, one can construct the doubled Hilbert space in order to build the thermal algebra and consequently simulate thermal qubits.

2.2. Thermo-SU(2) algebra

In order to double the SU(2) algebra, one needs to define the set of conjugation rules, which results in the following non-vanishing commutation rules

$$\begin{aligned}
&\left[S^{0}, S^{\pm}\right] = \pm S^{\pm} \\
&\left[S^{+}, S^{-}\right] = 2S_{0}, \\
&\left[\widetilde{S}^{0}, \widetilde{S}^{\pm}\right] = \pm \widetilde{S}^{\pm}, \\
&\left[\widetilde{S}^{+}, \widetilde{S}^{-}\right] = 2\widetilde{S}_{0}.
\end{aligned} \qquad (10)$$

Due to the fact that a unitary operator $U(\beta)$ can be assumed as a canonical transformation, for the algebra which characterizes the physical system, the so-called *TFD thermal operators* can be defined by

$$\begin{aligned}
S^{\pm}(\beta) &= U(\beta)S^{\pm}U^{-1}(\beta), \\
S^{0}(\beta) &= U(\beta)S^{0}U^{-1}(\beta), \\
\widetilde{S}^{\pm}(\beta) &= U(\beta)\widetilde{S}^{\pm}U^{-1}(\beta), \\
\widetilde{S}^{0}(\beta) &= U(\beta)\widetilde{S}^{0}U^{-1}(\beta),
\end{aligned} \qquad (11)$$

where $U(\beta)$ corresponds to the *Thermal Bogoliubov Transformation*. As a consequence, the thermal operators fulfill the following requirement of destroying the so-called thermal vacuum states $|0(\beta)\rangle$:

$$S^{-}(\beta)|0(\beta)\rangle = 0, \qquad \widetilde{S}^{-}(\beta)|0(\beta)\rangle = 0, \qquad (12)$$

where

$$\begin{aligned}
|0(\beta)\rangle &= U(\beta)|0,\widetilde{0}\rangle \\
&= \frac{1}{Z(\beta)^{1/2}} \sum_{n=0} e^{\beta\frac{n\omega}{2}}\left(S^{+}\widetilde{S}^{+} - S^{-}\widetilde{S}^{-}\right)^{n}|0,\widetilde{0}\rangle,
\end{aligned} \qquad (13)$$

with $Z(\beta) = 1 + e^{-\beta\omega}$ being the canonical partition function, and

$$U(\beta) = \frac{1}{Z(\beta)^{1/2}} \sum_{n=0} e^{-\beta\frac{n\omega}{2}}\left(S^{+}\widetilde{S}^{+} - S^{-}\widetilde{S}^{-}\right)^{n}. \quad (14)$$

The thermal vacuum states $|0(\beta)\rangle$ can be defined as the vacuum of the field emerging in a thermal bath, being an alternative way to represent the temperature in quantum mechanics in an algebraic perspective without the need for quantum statistics [27]. In this scenario, the Bogoliubov transformations can be seen as a rotation in the duplicated Hilbert space which leads to the creation of thermal effects [37, 38].

3. Thermal Qubits

Let us consider the particular case of a two-level system (qubit). From Eq. (13) one can define the thermal vacuum qubit or thermal qubit [14] as

$$|0(\beta)\rangle = \frac{1}{Z(\beta)^{1/2}}\left(|0,\widetilde{0}\rangle + e^{-\beta\frac{\omega}{2}}|1,\widetilde{1}\rangle\right). \qquad (15)$$

It is worth noting that, since the double thermofield is always in the same state as the real fermion, the mixed states $|1,\widetilde{0}\rangle$ and $|0,\widetilde{1}\rangle$ cannot be seen [27, 37, 38].

In this regard, rewriting Eq. (15) as

$$|0(\beta)\rangle = a|0,\widetilde{0}\rangle + b|1,\widetilde{1}\rangle, \qquad (16)$$

leads Eq. (16) to a form that resembles the geometrical representation for a pure state of a two-level quantum system, the so-called Bloch Sphere representation of a qubit [31]:

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle, \qquad (17)$$

where:

$$\begin{aligned}
a &= \cos\frac{\theta}{2} = \frac{1}{\sqrt{1+e^{-\beta\omega}}}, \\
b &= \sin\frac{\theta}{2} = \frac{e^{-\beta\omega/2}}{\sqrt{1+e^{-\beta\omega}}}.
\end{aligned} \qquad (18)$$

Thus, as a remarkable result, the thermal vacuum can be represented in a quantum circuit setup as sketched in Fig. 1. The circuit is created in order to implement the Bogoliubov transformation using the geometrical ideas of the Bloch sphere representation. The operator

$$R_{y}(\theta) = e^{-i\frac{\theta}{2}\sigma_{y}} = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \qquad (19)$$

represents a rotation of an angle θ around Y-direction on the non-tilde qubit $|0\rangle$, while the CNOT gate [31], controlled by the non-tilde qubit, targets the tilde one, doubling the Hilbert space in this way.

It is worth noting that this procedure is only possible due to the boson representation of SU(2) algebra reported previously in this section. This representation gives us a short circuit obtained through the analytical construction of the Bogoliubov transformation. Moreover, this result can be compared with the literature in the example reported in reference [23]. In this reference,
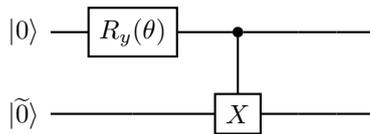
Figure 1: Sketch of the circuit that implements the *Thermal Bogoliubov Transformation* by applying the TFD approach for a single two-level quantum system. The state $|0\rangle$ is ground state in the $\mathcal{S}$ Hilbert space, while $|\widetilde{0}\rangle$ is the ground state of the $\widetilde{\mathcal{S}}$ one.

the authors use a function of IBM Qiskit [32] that decomposes a given unitary matrix into quantum gates. However, it currently only works for 2-qubit, while the circuit shown in Fig. 1 can be implemented as a subroutine limited only by the number of qubits present in a given quantum computer. In addition, using the Bogoliubov transformation approach also differs from other results reported recently in references [24], and [30], since it does not use a variational approach for simulating thermal qubits. Therefore, using the algebraic approach of TFD theory can make the simulation of quantum protocols at finite temperatures more intuitive for quantum algorithm developers, reducing the number of quantum gates needed for its implementation. This reduction of the number of gates can open a large avenue for the teaching of thermal effects on quantum protocols since it reduces the cost of implementing thermal qubits, simplifying the quantum logic in the circuits [28, 39, 40].

From Eq. (18), the angle θ in the Bloch sphere representation can be defined only by the reciprocal temperature β. Thus, one can conclude that

$$|0(\beta \to 0)\rangle = \frac{1}{\sqrt{2}}\Big(|0, \widetilde{0}\rangle + |1, \widetilde{1}\rangle\Big), \qquad (20)$$

$$|0(\beta \to \infty)\rangle = |0, \widetilde{0}\rangle, \qquad (21)$$

where in Eq. (20) we have the non-physical (infinity temperature) maximally thermal vacuum and Eq. (21) the non-thermal qubit. Therefore, increasing the temperature ($\beta \to 0$) will populate both ground and excited states equally, reaching the maximally thermal vacuum, Eq. (20). Consequently, decreasing it ($\beta \to \infty$) leads the system to the non-thermal state [23].

This result can be simulated by implementing the circuit sketched in Fig. 1 in a quantum computing setup using the IBM Qiskit framework [32]. Qiskit consists of a platform that enables high-level programming, serving as both a software development platform and a Quantum Computing language [2].

---

[2] For in-depth reading about QisKit, its initialization, commands and implementation, we recommend reading the references [32–34, 41–44].

3.1. Quantum algorithm

After installing a Python language compiler and *Qiskit* module[3], the readers are ready to learn how to write code to simulate their thermal qubits through TFD by constructing circuits and executing them on their own personal computers. To initiate the program, it is necessary to import the following modules into the Python environment:

`qiskit`: the main module, used for designing the quantum circuits and execute the quantum algorithms [43–45];

`qiskit.circuit`: to import the sequence of coherent quantum operations used in quantum computing. [43–45];

`numpy`: to create a mathematical workspace that involves multidimensional arrays and matrices, utilizing a vast range of mathematical functions available [46];

`matplotlib`: used for generating graphs or creating data visualizations in general [47];

`qiskit.tools.monitor`: used to monitor the real-time execution of our algorithms in a real quantum computer by utilizing the `job_monitor` function [43–45];

`qiskit.visualization`: used to visualize the probability distributions through the function `plot_histogram`.

This modules can be imported in the Python environment using the code:

Box 1: Importing the Packages

```python
from qiskit import *
from qiskit.circuit import *
import numpy as np
import matplotlib.pyplot as plt
from qiskit.tools.monitor import job_monitor
from qiskit.visualization import↵
↪plot_histogram
```

The next step is to define the rotation angle θ of the non-tilde qubit performed by the operator $R_y(\theta)$ in Eq. (19). Thus, we can represent the thermal vacuum in quantum circuit as

Box 2: Creating the Thermal Vacuum State

```python
# Define a variable theta to be a parameter↵
↪with name 'theta'
theta = Parameter('theta')
# Set number of qubits and classical bits↵
↪to 2
qubits_count = 2
bits_count = 2
```

---

[3] Tutorials available on the references [33, 42–45].

```
# Initialize a quantum circuit with two␣
 ↪qubits
qc = QuantumCircuit(qubits_count,bits_count)
# Add a parametrized RX rotation on the␣
 ↪qubit
qc.ry(theta,0)
qc.cx(0,1)
qc.draw(output='mpl')
```

The thermal vacuum state can be checked by measuring the corresponding qubits, the reader can measure the associated qubits and interpret the results using a classical computer. The outcomes of each qubit's measurement can be stored as classical bits (either 0 or 1) in the circuit's defined classical bits [33]. Qiskit offers a powerful solution for running quantum simulations on quantum processors using the cloud access to the IBM Quantum Experience platform [48], through the IBM Q Provider element [32, 42–44]. By specifying the number of circuit repetitions (or *shots*), the simulator obtain the counts for every measurement outcome in the final state. In order to utilize the IBM Q Experience through Qiskit and run projects on real quantum processors[4], it is necessary to create a free account. This can be done by accessing the "My Account" settings in IBM Q Experience [48]. Once the API token is obtained, IBM Q devices can be accessed from a home computer using Qiskit. To simulate the quantum circuit in the IBM quantum processor using a home-classical computer, use the following commands:

Box 3: Simulating the thermal vacuum state on a real quantum processor

```
#Adding the measurements in the circuit
qc.measure([0,1],[0,1])


#Saving your IBM account in your device
IBMQ.save_account('Users_Token')


#Executing the circuit in a real quantum␣
 ↪computer ibmq_lima
IBMQ.load_account() #Loading your IBM␣
 ↪Account
provider = IBMQ.get_provider(hub ='ibm-q')
qcomp = provider.get_backend('ibmq_lima')␣
 ↪#Selecting your quantum hardware
job = execute(qc, backend=qcomp,␣
 ↪shots=20000)
result = job.result()
counts = result.get_counts(qc)
```

---

[4] When implementing quantum logic gates, there can be systematic errors due to various factors such as noises and other processes. These inaccuracies in qubit control are not covered in this work. For more information on how these factors affect qubit control and quantum logic gates, refer to the Qiskit *User Guide* [44].
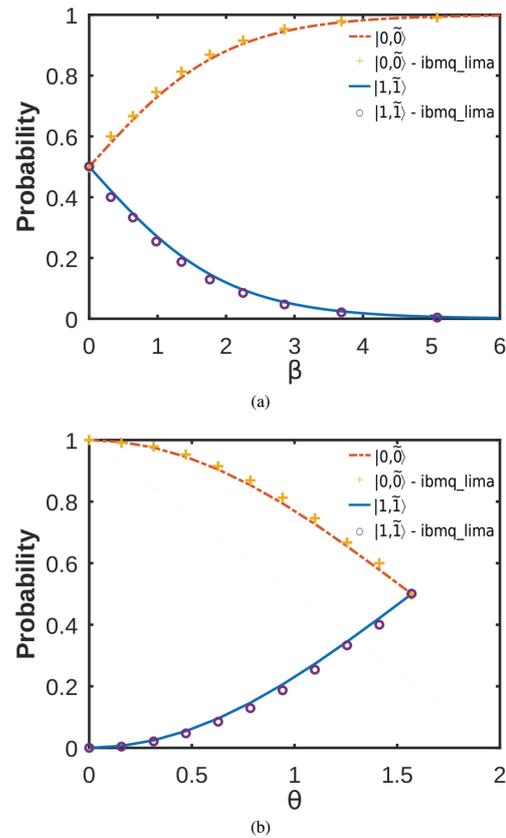
Figure 2: (a) Probabilities of the thermal vacuum state in terms of the reciprocal temperature β (dimensionless units). (b) Probability amplitude of the thermal vacuum in terms of θ (rad). Dashed (red) and solid (blue) lines represent the theoretical probabilities for the ground $|0,\widetilde{0}\rangle$ and excited $|1,\widetilde{1}\rangle$, respectively. They are obtained from the square modulus of the probability amplitudes a and b in Eq. (18).

Therefore, the thermal vacuum probabilities can be calculated by dividing the state counts by the number of shots. Fig. 2 shows the thermal vacuum probabilities as a function of the reciprocal temperature (β) and the angle θ. Dashed (red) and solid (blue) lines represent the theoretical probabilities for the ground $|0,\widetilde{0}\rangle$ and excited $|1,\widetilde{1}\rangle$, and are obtained from the square modulus of the probability amplitudes a and b in Eq. (18). The crosses and open circles represent the results obtained from the simulation of the thermal qubit in a real quantum processor, available in the backend *ibmq_lima*.

As can be seen in Fig. 2 (a), according to Eq.(21), thermal qubit at high temperature (low β) is a superposition of the states $|0,\widetilde{0}\rangle$ and $|1,\widetilde{1}\rangle$. This superposition decays exponentially until the qubit becomes the non-thermal state $|0,\widetilde{0}\rangle$ when the temperature drops (β rises), Eq. (20). In Fig. 2 (b) we have the same prediction of Fig. 2 (a), but now in terms of θ. It should be noted that $\theta \in [0, \pi/2]$ when $\beta \in [0, \infty]$, this result is in agreement with of quantum field theories at finite temperature that see β as a compactification of a time dimension [37, 38].

Similar results were recently shown in reference [23] in terms of β. However, the authors used a function *UnitaryGate* available in IBM Qiskit, which creates a quantum gate from a numeric unitary matrix [32]. On the other hand, the result showed in Fig. 2 is directly obtained from the Bloch sphere representation in terms of the Bogoliubov transformations, obtained from the circuit shown in Fig. 1, in which the angle of rotation in the $R_y(\theta)$ gate defines the temperature.

4. Quantum teleportation at finite temperature

In this regard, one can apply the formalism shown in this section in order to simulate a quantum algorithm involving thermofield states. As an example, we will simulate the quantum teleportation of thermofield states, a problem that has been previously discussed in reference [26]. In this reference, the quantum teleportation of thermal qubits will consist of the transmission of a non-thermal qubit $|0, \widetilde{0}\rangle$ into a thermal reservoir between two parties, conventionally known as Alice and Bob, spatially separated [26, 31]. Thus in the doubled Hilbert space representation, we have $|0, \widetilde{0}\rangle \rightarrow |0(\beta)\rangle$.

Initially, Alice has only a single copy of a non-thermal qubit, and she wants to send this qubit to Bob's thermal reservoir. Thus, Alice prepares the qubit whose information will be sent and uses a second (auxiliary) thermal vacuum state which will be maximally entangled to a third thermal vacuum state (auxiliary) belonging to Bob, who will receive Alice's non-thermal qubit. In the following, Alice performs measurements on its two thermal qubits and informs her results through a classical channel to Bob. Finally, with this (classical) information, Bob properly performs a set of (quantum) operations in your thermal vacuum state to retrieve the non-thermal qubit sent by Alice in his thermal reservoir [26]. Therefore, at the end of the teleportation, the initial non-thermal qubit $|0, \widetilde{0}\rangle$ acquires the temperature information of Bob's thermal reservoir, becoming the thermal qubit $|0(\beta)\rangle$ in the doubled Hilbert space representation of the TFD approach, Eq. (15).

From this protocol, one can build the circuit in IBM Qiskit that simulates this procedure, as shown in Fig. 3. The algorithm is divided into five main steps, described as follows.

(I) **Preparing the thermal vacuum states**. Since the usual teleport algorithm uses three qubits [31], the TFD's version needs six qubits in order to implement Bogoliubov's transformations. The first four qubits $\{q_1, \tilde{q}_1, q_2, \tilde{q}_2\}$ belongs to Alice, while the last two $\{q_3, \tilde{q}_3\}$ belongs to Bob. Thus, the first step of the circuit (after calling the packages as presented in Box 1) is to create the circuit registers:

Box 4: Creating the circuit register for the TFD teleportation algorithm

```
# Creating registers
qreg_q = QuantumRegister(6, 'q')
creg_c = ClassicalRegister(2, 'c')
circuit = QuantumCircuit(qreg_q, creg_c)
```

In addition, Alice's first two qubits $\{q_0, \tilde{q}_0\}$ are prepared on a non-thermal state in the doubled Hilbert space using the general `U-gate` defined as applied in the tilde and non-tilde qubits:

$$\mathcal{U}(\alpha, \beta, \gamma) = \begin{pmatrix} \cos\frac{\alpha}{2} & -e^{\gamma}\sin\frac{\alpha}{2} \\ e^{\beta}\sin\frac{\alpha}{2} & e^{\beta+\gamma}\cos\frac{\alpha}{2} \end{pmatrix} \qquad (22)$$

Box 5 displays the code used to initialize Alice's non-thermal state.:

Box 5: Initialization of Alice's non-thermal state

```
#Alice prepares her non-thermal state to be␣
 ↪teleported
circuit.u(alpha,beta,gamma,qreg_q[0])
circuit.u(alpha,beta,gamma,qreg_q[1])
```

The parameters `alpha`, `beta` and `gamma` on Box 5 can be selected by the user in order to initialize the state to be teleported.

On the other hand, using the circuit shown in Fig. 1 (code shown in Box 2), Alice prepares an auxiliary thermal vacuum state $|0(\beta)\rangle_A$ in qubits $\{q2, \tilde{q_2}\}$ while Bob does the same in qubits $\{q3, \tilde{q_3}\}$, preparing the state $|0(\beta)\rangle_B$. The following Box shows the code for this initialization of Alice's and Bob's thermal vacuum states:

Box 6: Initialization of Alice's and Bob's thermal vacuum states

```
#Alice and Bob prepares their thermal␣
 ↪vacuum states
circuit.ry(theta, qreg_q[2])
circuit.cx(qreg_q[2], qreg_q[3])
circuit.ry(theta, qreg_q[4])
circuit.cx(qreg_q[4], qreg_q[5])
circuit.barrier()
```

The user can define the temperature of the reservoir by selecting the value of parameter `theta` using Eq. (18).

(II) **Preparing the entanglement with the auxiliary thermal vacuum states**. The next step is to entangle Alice's auxiliary vacuum state $|0(\beta)\rangle_A$ with Bob's $|0(\beta)\rangle_B$, in the doubled Hilbert Space. Box 7 shows the code that for the entanglement of the thermal

vacuum states $|0(\beta)\rangle_A$ and $|0(\beta)\rangle_B$:

Box 7: Generating entanglement between Alice and Bob's thermal vacuum states

```
#Preparing the entanglement between Alice␣
 ↪and Bob's thermal vacuum states
circuit.h(qreg_q[2])
circuit.cx(qreg_q[2], qreg_q[4])
circuit.h(qreg_q[3])
circuit.cx(qreg_q[3], qreg_q[5])
circuit.barrier()
```

(III) **Sending the non-thermal qubit**. Then Alice starts the process of sending the non-thermal state using his auxiliary thermal vacuum state $|0(\beta)\rangle_A$, which is now entangled with Bob's $|0(\beta)\rangle_B$. The code for this process is displayed in the following box:

Box 8: Decoding Alice's non-thermal state into her auxiliary thermal vacuum state.

```
#Alice decodes her non-termal state into␣
 ↪his auxiliary thermal vacuum state
circuit.cx(qreg_q[0], qreg_q[2])
circuit.h(qreg_q[0])
circuit.cx(qreg_q[1], qreg_q[3])
circuit.h(qreg_q[1])
circuit.barrier()
```

(IV) **Conditioned operations**. The following step would be for Alice to perform measurements on her qubits. Depending on the results, contact Bob through a classic channel to inform the corrections that Bob must apply in his state in order to finish the teleportation process, and he can retrieve the non-thermal qubit sent by Alice. This step can be performed through an operation conditioned to the result of Alice's measurements. However, IBM QE does not allow the implementation of this type of port conditioned to a classic channel [32]. In this case, we can replace it with the `CNOT` and `Z-Controlled` gates. In this regard, Alice performs

controlled gates on Bob's thermal vacuum state using her qubits as controls. Thus, we were able to modify the original circuit without changing its main objective. The following Box displays the code for this process:

Box 9: Executing Alice's conditioned gates on Bob's thermal vacuum state.

```
#Alice performs conditioned gates targeted␣
 ↪to Bob's thermal vacuum state
circuit.cx(qreg_q[2], qreg_q[4])
circuit.cz(qreg_q[0], qreg_q[4])
circuit.cx(qreg_q[3], qreg_q[5])
circuit.cz(qreg_q[1], qreg_q[5])
circuit.barrier()
```

(V) **Measurement in Bob's vacuum state**. Finally, Bob measures his thermal vacuum state, retrieving the non-thermal qubit sent by Alice into his thermal reservoir as the thermal qubit $|0(\beta)\rangle$, Eq. (15). The following Box exhibits the code for this step:

Box 10: Measuring Bob's thermal vacuum state.

```
# Bob's measurements
circuit.measure(qreg_q[4], creg_c[0])
circuit.measure(qreg_q[5], creg_c[1])
circuit.draw(output='mpl')
```

Thus, one can simulate the teleport algorithm through TFD algebraic perspective (Fig. 3) using IBM Qiskit. For the sake of simplicity, we chose the angles $\{\alpha = \pi, \beta = 0, \gamma = \pi\}$ in Eq. (22), during step I (Box 4). Thus, Alice sends the non-thermal state $|1, \tilde{1}\rangle$. Moreover, considering the microwave frequency $\hbar\omega = 1.5 \times 10^{-22}$ J in Eq. (18), one can prepare the thermal vacuum states $|0(\beta)\rangle_A$ and $|0(\beta)\rangle_B$ at the finite temperature 72 K, by selecting the parameter $\theta = \pi/3$ in the step I (Box 5). In this regard, one can simulate quantum teleportation at a finite temperature (Fig. 3) in both ideal and real quantum processors [33]. Qiskit provides a Quantum Assembly Language (QASM) simulator for
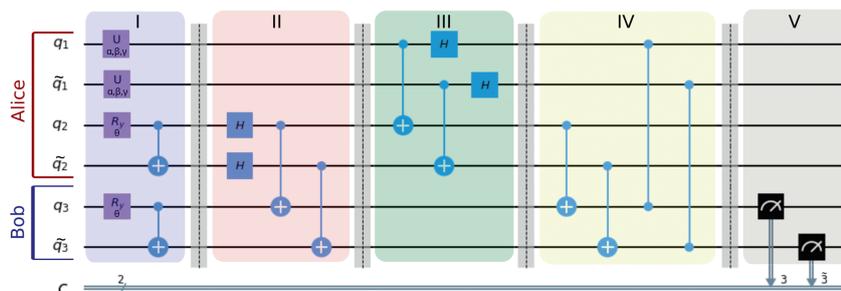

Figure 3: Teleportation circuit of a non-thermal qubit to a thermal vacuum state. The first four qubits $\{q_0, q_1, q_2, q_3\}$ belongs to Alice, while the last two $\{q_4, q_5\}$ belongs to Bob. The protocol can be divided into five main steps: (I) Preparing the thermal vacuum states; (II) Preparing the entanglement with Alice's auxiliary vacuum state $|0(\beta)\rangle_A$ and Bob's $|0(\beta)\rangle_B$; (III) Sending the non-thermal qubit; (IV) Conditioned operations targeting in Bob's state; (V) Measurement of Bob's state in order to retrieve the non-thermal qubit sent by Alice.

quantum simulation on personal computers [32, 42–44]. This tool is an integral part of the module and is specifically designed to emulate the execution of quantum circuits on a local classical processor. This enables us to emulate an ideal quantum processor without any external environmental disturbances. On the other hand, in addition to being able to simulate our quantum circuit on a numerically emulated ideal processor on a home-classical computer, we can also run our projects on real quantum processors using the IBM Q Experience through the IBM Q Provider element (as shown in Box 3). In the following, Box 11 shows the code for running the quantum teleportation at a finite temperature (Fig. 3) in both ideal and real quantum processors:

Box 11: Simulating quantum teleportation under finite temperature using an ideal and a real quantum processor.

```
#Ideal quantum processor
simulator = Aer.
 ↪get_backend('qasm_simulator')
result = execute(circuit,␣
 ↪backend=simulator, shots = 20000).result()
plot_histogram(result.get_counts(circuit))

#Real quantum processor ibm_lagos
IBMQ.load_account() #Loading your IBM␣
 ↪Account
provider = IBMQ.get_provider(hub ='ibm-q')
qcomp = provider.get_backend('ibm_lagos')␣
 ↪#Selecting your quantum hardware
job = execute(circuit, backend=qcomp,␣
 ↪shots=20000)
result = job.result()
counts = result.get_counts(circuit)
```

Fig. 4 shows the probability distributions for this teleport algorithm simulated in ideal quantum processor (`QASM` simulator [49]) and the 7-qubit real quantum processor `ibm_lagos`[5]. As can be seen, the state measured in Bob's vacuum state $|0(\beta)\rangle_B$ was, in a good approximation, the thermal qubit state

$$|1(\beta)\rangle = \sqrt{\frac{1}{4}}|0,\tilde{0}\rangle + \sqrt{\frac{3}{4}}|1,\tilde{1}\rangle , \qquad (23)$$

which is in agreement with the application of the Thermal Bogoliubov Transformation, Eq. (14), on Alice's non-thermal state $|1,\tilde{1}\rangle$, with corresponding temperature 72 K. Thus, the non-thermal state sent by Alice acquires the temperature information of the bath, becoming a thermal qubit in agreement with the presented Thermofield Dynamics model.

---

[5] The `ibm_lagos` processor architecture can be found on IBM QE platform [48]. The details of the processor are beyond the scope of this paper.
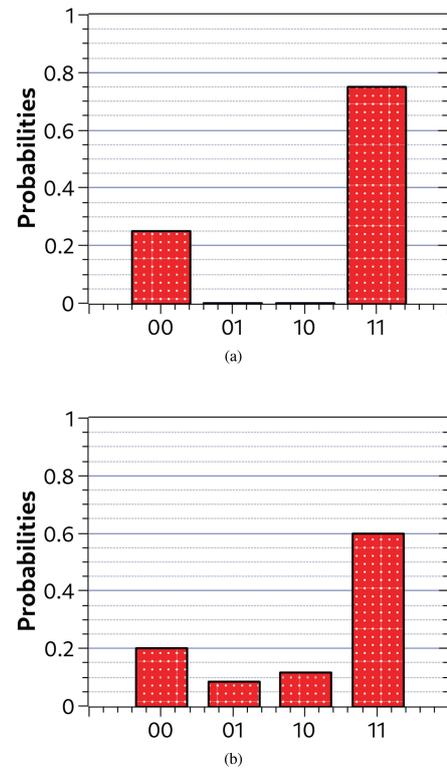
Figure 4: Probability distribution for the teleportation algorithm at finite temperature simulated for 20000 shots in (a) an ideal quantum processor (QASM simulator [49]), and (b) a 7-qubit real quantum processor `ibm_lagos`. The simulation is performed considering the frequency $\hbar\omega = 1.5 \times 10^{-22}$ J and 72 K for the temperature of the thermal vacuum states.

The ideal processor numerically emulates the quantum circuit execution shown in Fig. 3(a), without any influence from external perturbations due to the inevitable coupling between quantum information processing systems and the external environment. The simulation returns the counts of each measurement in the final thermal state for the given set of 20000 shots. However, even if no decoherence act and no additional errors affect the system statistics, due to the fact that the number of shots is not large enough, one can still notice a slight statistical discrepancy between the obtained result and the theoretically expected result observed in Fig. 2(b). On the other hand, in the real quantum processor, noise and other decohering processes lead to imprecision in the fine control of the qubits, causing systematic errors in the implementation of logic gates, which explains the appearance of the states $|0,\tilde{1}\rangle$ and $|1,\tilde{0}\rangle$. Nevertheless, these results show a proof of concept regarding the implementation of thermal qubits through the TFD approach. Therefore, using the Bloch sphere representation in terms of the Bogoliubov transformation allows the construction of thermal qubits in a quantum computing setup, opening a large avenue for future research toward the study of thermal effects in the development of optimal quantum protocols.

5. Conclusion

In summary, this work shows how quantum computing can be a useful tool for teaching the Thermofield Dynamics and the Blogoliobov transformations in a practical way. We show a didactic implementation of thermal vacuum qubits in a quantum computing setup, using an algebraic approach accessible to most undergraduate physics students. The Bloch sphere representation for a qubit is built in terms of the Bogoliubov transformations, and the thermofield-double space is simulated through the IBM Qiskit. The approach reported in this work can be implemented as a practical algebraic framework in undergraduate and graduate classrooms to study the thermal effects of the TFD approach using quantum computing. The presented examples show that constructing thermal qubits using TFD in a quantum computing setup can be more intuitive and approachable for professors, students, and even quantum algorithm developers. The presented method appears as an alternative to the complicated variational approach methods used in the literature up to date [24, 30], reducing the number of quantum gates and processes needed for the implementation of the thermal qubits. In addition, the use of TFD theory for quantum computing opens a large avenue for physics teaching on the thermal effects on quantum protocols, which can be a path toward the understanding of the cost of implementation of thermal quantum algorithms. Therefore, this work highlights the potential of TFD algebraic perspective in quantum computing to encourage the study of optimal quantum protocols.

Acknowledgements

References

[1] M.R. Wasielewski, M.D. Forbes, N.L. Frank, K. Kowalski, G.D. Scholes, J. Yuen-Zhou, M.A. Baldo, D.E. Freedman, R.H. Goldsmith, T. Goodson et al., Nature Reviews Chemistry **1**, 490 (2020).

[2] M.A. Ponte and A.C. Santos, Sci. Rep. **9**, 10470 (2019).

[3] A.C. Santos, A. Saguia and M.S. Sarandy, Phys. Rev. E **101**, 062114 (2020).

[4] G.M. Andolina, M. Keck, A. Mari, M. Campisi, V. Giovannetti and M. Polini, Phys. Rev. Lett. **122**, 047702 (2019).

[5] B. Çakmak, arXiv:2005.08489 (2020).

[6] A.C. Santos, B. Çakmak, S. Campbell and N.T. Zinner, Phys. Rev. E **100**, 032107 (2019).

[7] A. Gaita-Ariño, F. Luis, S. Hill and E. Coronado, Nature chemistry **11**, 301 (2019).

[8] Y.A. Mezenov, A.A. Krasilin, V.P. Dzyuba, A. Nominé and V.A. Milichko, Advanced Science **6**, 1900506 (2019).

[9] C. Cruz, M.F. Anka, M.S. Reis, R. Bachelard and A.C. Santos, Quantum Science and Technology **7**, 025020 (2022).

[10] A. Auffèves, PRX Quantum **3**, 020101 (2022).

[11] M. Schlosshauer, Physics Reports **831**, 1 (2019).

[12] C. Cruz, D.O. Soares-Pinto, P. Brandão, A.M. Santos and M.S. Reis, EPL (Europhysics Letters) **113**, 40004 (2016).

[13] C. Cruz and M. Anka, EPL (Europhysics Letters) **130**, 30006 (2020).

[14] H. Xu, Y.C. Ong and M.H. Yung, Physical Review A **105**, 012430 (2022).

[15] D. Tamascelli, A. Smirne, J. Lim, S.F. Huelga and M.B. Plenio, Phys. Rev. Lett. **123**, 090402 (2019).

[16] G. Harsha, T.M. Henderson and G.E. Scuseria, The Journal of Chemical Physics **150**, 154109 (2019).

[17] C.K. Lee, S.X. Zhang, C.Y. Hsieh, S. Zhang and L. Shi, arXiv:2206.05571 (2022).

[18] R. Cartas-Fuentevilla, J. Berra-Montiel and O. Meza-Aldama, The European Physical Journal C **80**, 1 (2020).

[19] Z. Xu, A. Chenu, T. Prosen and A. del Campo, Physical Review B **103**, 064309 (2021).

[20] K. Mallayya, M. Rigol and W. De Roeck, Phys. Rev. X **9**, 021027 (2019).

[21] N. Lopes, D.G. Barci and M.A. Continentino, Journal of Physics: Condensed Matter **32**, 415601 (2020).

[22] P. Weber, V. Lula-Rocha, J. Pereira, M. Trindade, L. Silva Filho, M. Martins, A. Santana and J. Vianna, Annals of Physics **443**, 168986 (2022).

[23] R. Miceli and M. McGuigan, in: *2019 New York Scientific Data Summit (NYSDS)* (New York, 2019).

[24] R. Sagastizabal, S. Premaratne, B. Klaver, M. Rol, V. Negîrneac, M. Moreira, X. Zou, S. Johri, N. Muthusubramanian, M. Beekman et al., npj Quantum Information **7**, 130 (2021).

[25] D. Zhu, S. Johri, N.M. Linke, K. Landsman, C.H. Alderete, N.H. Nguyen, A. Matsuura, T. Hsieh and C. Monroe, Proceedings of the National Academy of Sciences **117**, 25402 (2020).

[26] T. Prudêncio, T. Rocha Filho and A. Santana, Physica Scripta **94**, 095102 (2019).

[27] A.E. Santana, J. Malbouisson, A. Malbouisson and F. Khanna, in: *Non-linear Dynamics and Fundamental Interactions*, edited by F. Khanna and D. Matrasulov (Springer Nature, Berlin 2006).

[28] B.P. Lanyon, M. Barbieri, M.P. Almeida, T. Jennewein, T.C. Ralph, K.J. Resch, G.J. Pryde, J.L. O'brien, A. Gilchrist and A.G. White, Nature Physics **5**, 134 (2009).

[29] D.A. Rowlands and A. Lamacraft, Physical Review B **98**, 195125 (2018).

[30] J. Wu and T.H. Hsieh, Physical review letters **123**, 220502 (2019).

[31] M.A. Nielsen and I. Chuang, Quantum computation and quantum information (American Association of Physics Teachers, Maryland, 2002).

[32] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F.J. Cabrera-Hernández, J. Carballo-Franquis, A. Chen, C.F. Chen, J.M. Chow et al., *Qiskit: An open-source framework for quantum computing*, available in: https://zenodo.org/doi/10.5281/zenodo.2562110

[33] G.F. Jesus, M.H.F. Silva, T.G.D. Netto, L.Q. Galvão, F.G.O. Souza and C. Cruz, Revista Brasileira de Ensino de Física **43**, e20210033 (2021).

[34] W.M.S. Alves and J.C.C. Felipe, Revista Brasileira de Ensino de Física **44**, e20210290 (2022).

[35] A.N. Oliveira, E.V. Oliveira, A.C. Santos and C.J. Villas-Boas, Revista Brasileira de Ensino de Física **44**, e20210333 (2021)

[36] G. Petronilo, *Simulating thermal qubits throught thermofield dynamics*, available in: https://github.com/GustavoPetronilo/GustavoPetronilo/blob/Quantum-Computing/Simulating_thermal_qubits.ipynb, accessed in: 18/09/2023.

[37] H. Umezawa, *Advanced field theory: Micro, macro, and thermal physics* (American Institute of Physics, Maryland, 1993).

[38] F. Khanna and D. Matrasulov, *Non-Linear Dynamics and Fundamental Interactions* (Springer Science & Business Media, Berlin, 2005), v. 213.

[39] R. Wille, M. Soeken and R. Drechsler, in: *Design Automation Conference* (Anaheim, 2010).

[40] B. Langenberg, H. Pham and R. Steinwandt, IEEE Transactions on Quantum Engineering **1**, 2500112 (2020).

[41] H. Norlen, *Quantum Computing in Practice with Qiskit® and IBM Quantum Experience®* (Packt Publishing, Mumbai, 2020).

[42] Qiskit, *Qiskit 0.45 documentation*, available in: https://qiskit.org/documentation, accessed in: 29/08/2023.

[43] Qiskit, *Qiskit Textbook*, available in: https://qiskit.org/textbook, accessed in: 28/08/2023.

[44] GitHub, *qiskit-tutorials*, available in: https://github.com/Qiskit/qiskit-tutorials, accessed in: 15/01/2023.

[45] Qiskit, available in: https://github.com/Qiskit, accessed in: 28/08/2023.

[46] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith et al., Nature **585**, 357 (2020).

[47] J.D. Hunter, Computing in Science & Engineering **9**, 90 (2007).

[48] IBM Quantum, available in: https://quantum-computing.ibm.com, accessed: in 18/09/2023.

[49] D.C. McKay, T. Alexander, L. Bello, M.J. Biercuk, L. Bishop, J. Chen, J.M. Chow, A.D.Córcoles, D. Egger, S. Filipp et al., arXiv:1809.03452 (2018).