Revista Brasileira de Ensino de Física, vol. 42, e20190115 (2020)
www.scielo.br/rbef
DOI: http://dx.doi.org/10.1590/1806-9126-RBEF-2019-0115

Classical simulation of Grovers quantum algorithm

Jairo Ernesto Castillo*[1], Yesenia Sierra[1], Nelson L. Cubillos[1]

[1]Univesidad Distrital Francisco Jóse de Caldas , Bogotá, Colombia

Received on May 16, 2019. Revised on August 21, 2019. Accepted on August 24, 2019.

This paper in a didactic way, presents an introduction to quantum computing where some quantum formalisms are analyzed to finally address Grover's algorithm. It is widely known that this algorithm is one of the key algorithms in quantum computing, being its ability to explode successfully the superposition principle one of the reasons. In addition, this algorithm can be used for both locating efficiently a specific element in a cluttered database and solving certain problems when it is difficult finding a proper solution, but at the same time it is very simple to try with possible candidates. Finally, a simulation of this algorithm is carried out and the results are compared with other classical algorithms to illustrate the significant potential advantages of quantum computing.
Keywords: quantun computing, Grovers algorithm, simulation

1. Introduction

It is well known that since the mid-nineteenth century with the emergence of classical computing under the architecture of the Church-Turing and Von Neumann machines [1], it was possible to build a computer capable of performing basic operations such as addition and multiplication in a few seconds. Over the years, technological developments have allowed optimizing the response time of all kinds of operations; however, there are still a large number of mathematical calculations that would take decades to be solved (e.g., determining whether a large number is prime or not.). These types of complex calculations encouraged Paul Beniof [2, 3, 4] and Richard Feynmann [5] to reconsider the idea of using computers under the principles of quantum physics, so they started working with classical computers that they adapted and operated with some of the main principles of quantum mechanics. On the other hand, every day it becomes more difficult to handle classical computing because of the constant demands of miniaturization that electronic components require, knowing that soon it will not be possible to reduce them anymore and the laws of classical physics will not be considered anymore. An example of this is that transistors have a limit where they simply stop working properly. This limit is met when the electrons escape from the channels where they are supposed to flow because of the so-called "tunnel effect", that in other words means if a classical particle reaches an obstacle, normally this particle can go through it and bounces back, but the electrons are quantum particles which exhibit wave behavior keeping the possibility that a significant portion of these electrons may go through the walls where they are confined. In this way, the signal can interfere through channels where it is not supposed to, causing the chip to stop working [6]; therefore, at some point, the technical progress will come to an end leaving space to the development of quantum mechanics.

It is important to stress that a quantum computer can perform all kinds of tasks that a classical computer does thanks to the fact that computability is the same, the difference lies in the time of convergence in the results obtained by the running algorithms since quantum computers have the ability to process information in parallel in a massive way. This is known as the superposition principle or superposition property; superposition is the fundamental law of quantum mechanics which defines the collection of all possible states that an object can have, meaning that a system can even be in two or more of its states at the same time, which allows the development of new and innovating algorithms.

Grover's algorithm is one of the most important algorithms of the quantum computing illustrating the fundamental quantum superposition principle by searching N records in an unordered sequence inspecting all possible combinations simultaneously, causing a significant reduction of the response time to $O(\sqrt{N})$ steps compared with a classical algorithm. Grover's algorithm is defined as a probabilistic algorithm, so the correct answer is obtained with a certain error probability that, at the same time, can be as low as desired by running more iterations. This algorithm receives a list of numbers and simultaneously the specific data to be found; afterward, a function f which its primary goal is to assign the state $f(x) = 0$ if the element that is being searched is not requested; otherwise, the assigned state is $f(x) = 1$. As soon as the desired data matches a record, the amplitude of the mentioned state is modified without affecting the other

*Correspondence email address: jairocastillo63@yahoo.es

elements, obtaining that in the end, the probability of one state differs completely from the rest. The simplicity and helpfulness of this algorithm stand out the speed on response times intended to implement in the computers of the future.

It is important to note that a classic simulation of the Grover algorithm and the study of other quantum algorithms has been studied in different works such as [7,8,9,10,11].

Unlike these works, in this article, in a didactic way and sequentially, the physical and mathematical foundations are constructed to better understand the Grover algorithm in three steps.

For the simulation of the algorithm the Java programming language is used, because it is independent of the platform and because it has no problems with the release of the system memory. The implementation of the algorithm is easy to understand and is according to previously developed concepts.

This paper has been organized in the following way: First of all, preliminary concepts are introduced and some concepts of the formalisms of quantum mechanics are illustrated which will allow introducing Grover's algorithm that will be covered in the second part. Last of all, a classical simulation of Grover's algorithm is carried out and the results are compared with other classical algorithms.

2. Preliminary concepts

2.1. About the superposition principle

Following Penrose [12], "think of the position of an electron." As we know, in a classical image, the electron by an indivisible particle could be located in position A or position B, however in the mechanical quantum-image, in some sense it has the possibility of occupying both positions simultaneously. Let $\mid A\rangle$ be the state for the electron in position A and be $\mid B\rangle$ the state of the electron in position B. According to the quantum theory (principle of superposition), there are other possible states open to the electron that is written as $w \mid A\rangle + z \mid B\rangle$ where w and z are complex weight factors. If the weight factors are taken as real numbers (not negative), then it could be considered that this combination represents, in a certain sense, a weighted probability of the expected value for the position of the electron, where w and z represent the weighted probabilities of the electron in the position A or in the position B. If $w = 0$ then the electron would certainly be in B, and if $z = 0$ it would be in A. If $w = z = 1$, then it represents equal possibilities for the electron in A or in B. But w and z are Complex, so that such an interpretation does not make any sense. As Penrose [12] states, we can not say, in familiar and everyday terms, what "means" that an electron is in a state of superposition of two places at the same time with complex w and z weights. For the moment, we must simply accept that this is really the kind of description we have to adopt for quantum-level systems. In fact, quantum superposition gives rise to many directly observable effects, such as the interference peaks of a wave electron in the double-slit experiment, the last of them made in 2008 [13] its most ambitious form, electron to electron, thus proving the quantum-mechanical hypothesis

2.2. Quantum bits and Unitary transformations.

In classical computation the minimum unit of information is the bit, which can be a 0 or 1 state. A quantum bit or qubit can be in a state that is a superposition of states 0 and 1. In mathematical formalism a qubit is a Hilbert space vector H^2

Definition 2.1 A qubit or quantum bit is a normalized vector of the space of H^2 .Considering the base $\{|0\rangle, |1\rangle\}$ of H^2 , any qubit can be written as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, with $|\alpha|^2 + |\beta|^2 = 1$

Definition 2.2 A system of n-qubits is a space vector $H^{2^n} = \otimes_{i=1}^{n} H^2$.

Definition 2.3 A quantum algorithm consists of the evolution of a system represented by n-qubits.

To describe the states of a quantum system, the Dirac representation and notation is usually used [9].

Instead of writing the vector as $\overrightarrow{v}$ the notation ket $|v\rangle$ is used. In particular the base $\{|0\rangle, |1\rangle\}$ of H^2 is written as:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1}$$

Therefore, in H^2 any vector representing the state of a quantum system is written as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{2}$$

Definition 2.4 Every vector ket has the form:

$$|\psi\rangle = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ . \\ . \\ . \\ \alpha_N \end{pmatrix} \tag{3}$$

And bra a vector of the form:

$$\langle\psi| = (\alpha_1^*, \alpha_2^*, ...\alpha_N^*) \tag{4}$$

Definition 2.5 the scalar product of bras and kets, we call it "braket":

$$\langle\psi|\varphi\rangle = (\alpha_1^*, \alpha_2^*, ...\alpha_N^*) \begin{pmatrix} \beta_1 \\ \beta_2 \\ . \\ . \\ . \\ \beta_N \end{pmatrix} = a\epsilon\mathbb{C} \quad (5)$$

Definition 2.6 Given a set $B = \{|u_1\rangle, |u_2\rangle, ...|u_N\rangle\}$, B is an orthonormal basis of H^N if for all i, j we have $\langle u_i|u_j\rangle = \delta_{i,j} = \{^{1 \text{ si } i=j}_{0 \text{ si } i\neq j}$

Theorem 2.1 Let $B = \{|u_1\rangle, |u_2\rangle, ...|u_N\rangle\}$ be an orthonormal basis, then $\sum_{i=1}^{N} |u_i\rangle\langle u_i| = I$

Demonstration.

As $I|\psi\rangle = (\sum_{i=1}^{N}|u_i\rangle\langle u_i|)(\sum_{j=1}^{N} a_j|u_j\rangle) = \sum_{i=1}^{N}\sum_{j=1}^{N} a_j|u_i\rangle\langle u_i|u_j\rangle = \sum_{i=1}^{N} a_i|u_i\rangle = |\psi\rangle$

Definition 2.7 An operator A of H^N is a square matrix of dimension N and complex coefficients.

Definition 2.8 The tensor product $\otimes$, also called the external product between two matrices A and B , is defined as the matrix.

$$C = A \otimes B = \begin{pmatrix} a_{11}B & .. & .. & a_{1m}B \\ . & & & . \\ . & & & . \\ a_{n1}B & .. & .. & a_{nm}B \end{pmatrix} \quad (6)$$

Definition 2.9 The adjunct of an operator A is denoted by $A^\dagger$ and is defined as the transposed and conjugated operator of A. That is, if $\alpha_{ij} = \langle u_i|A|u_j\rangle$ are the components of A, the components of $A^\dagger$, are $\alpha^*_{ji} = \langle u_j|A|u_i\rangle^* = \langle u_i|A^\dagger|u_j\rangle$

Definition 2.10 An operator A is Hermitian if $A = A^\dagger$. If it is Hermitian its diagonal must be real, since $\alpha_{ij} = \alpha^*_{ji}$, therefore $\alpha_{ii} = \alpha^*_{ii}$,that is, their eigenvalues are real.

Definition 2.11 An operator U is unitary if $U^\dagger U = UU^\dagger = I$

Definition 2.12 To the operators of the form $P = |\varphi\rangle\langle\varphi|$ they are called projectors,since they project orthogonally any ket $|\Phi\rangle$ on a ket $|\varphi\rangle$

$$P|\Phi\rangle = |\varphi\rangle\langle\varphi|\Phi\rangle = a|\varphi\rangle \quad (7)$$

Definition 2.13 A set of projectors $\{M_1, M_2.....M_l\}$ is said to be a measurement operator if

$$\sum_{i=1}^{l} M_i M_i^\dagger = I \quad (8)$$

Definition 2.14 Unitary and Hermitian operators are called quantum gates.

Definition 2.15 It is said that a system represented by a ket $|\psi\rangle$ evolves into system $|\varphi\rangle$ when one of the following operations is carried out:

$$|\psi\rangle \overset{U}{\to} |\varphi\rangle o|\psi\rangle \overset{M}{\to} |\varphi\rangle \quad (9)$$

Definition 2.16 Quantum gates.The most important quantum gates, for their usefulness in the design of algorithms, are the following:

Gate h of Walsh-**Hadamard**.

The Hadamard gate applied to $n-$qubits, is known as the Walsh-Hadamard transformation and produces the superposition off all 2^n possible states.

This transformation is defined as:

$$W = h \otimes h \otimes ..... \otimes h \quad (10)$$

Where $h = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. For example, the transformation of WalshHadamard applied to $n = 2$ qubits is defined as:

$$\begin{aligned} W &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ &= \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \end{aligned} \quad (11)$$

3. Grover's algorithm

This algorithm is one of the most important of quantum computing since it exploits the principle of superposition to the maximum. It is a very attractive algorithm, since it can be used both to efficiently locate a certain element in a disorganized database, and to solve those problems in which it is very difficult to find a solution, but at the same time very simple tot est possible candidates. That is, a list of size N is available. The classical theory needs, on average, to read $\frac{N}{2}$ values, however, the Grover algorithm requires only $\sqrt{N}$ interactions.

Mathematically the problem can be reduced to finding a $x \in \{0, 1, 2, ...N-1\}$ with $N = 2^n$, such that $f(x) = a$ for a a known. The idea proposed by Grover is based on the implementation of a function f such that,if w denoted the x sought, then:

$$f(x) = \{ \begin{matrix} 0 \text{ if } x \neq w \\ 1 \text{ if } x = w \end{matrix} \quad (12)$$

From the perspective of a quantum computer, what is needed is a unitary transformationU_f such that $U_f|x\rangle = (-1)^{f(x)}|x\rangle = \{ \begin{matrix} |x\rangle \text{ if } x \neq w \\ -|x\rangle \text{ if } x = w \end{matrix}$

As we can observe in the previous equation, if it is a state x which is not the mark, then the transformation will have no effect on the state $|x\rangle$. If now

$U_f|w\rangle = (-1)^{f(w)}|w\rangle$ then the result will be $U_f|w\rangle = -|w\rangle$. The geometric interpretation is that the unitary matrix U_f makes a reflection in its amplitude to the marked within the set of N

elements.

Grover's algorithm works in three steps as follows:

Step No1. It consists in the normalization of the amplitudes of all the states, which is achieved with a transformation of Hadamard: $|s\rangle = h^{\otimes n}|0\rangle$,therefore,in the instant $t = 0$, $|\psi_{t=0}\rangle = |\psi_0\rangle = |s\rangle$,that is, all states have the same amplitude of probability.

Step No2. The reflection U_f is applied to all states: $|\psi_t\rangle = U_f|\psi_0\rangle$ where if $|\psi_t\rangle$ is the state $|w\rangle$, then it can be interpreted as a reflection negative $-|w\rangle$ and yes they are the non-marked states $|x\rangle$ the transformation has not effect. Mathematically the transformation U_f has the following form:

$$U_f = I - 2|w\rangle\langle w| \tag{13}$$

It is not difficult to observe that: $U_f|w\rangle = (I - 2|w\rangle\langle w|)|w\rangle = I|w\rangle - 2|w\rangle\langle w|w\rangle = |w\rangle - 2|w\rangle = -|w\rangle$

and that $U_f|x\rangle = (I-2|w\rangle\langle w|)|x\rangle = I|x\rangle-2|w\rangle\langle w|x\rangle = I|x\rangle = |x\rangle$ since $\langle w|x\rangle = 0$.

The transformation (13) could classically be interpreted as an If-then-else selection operator.

Step No3. The third step is to apply a reflection to the state $|s\rangle$, which is described as:

$$U_s = 2|s\rangle\langle s| - I \tag{14}$$

The transformation (14) modifies all the states to $U_s|\psi_t\rangle$ and thus completes the transformation:

$$|\psi_{t+1}\rangle = U_s U_f|\psi_0\rangle \tag{15}$$

The transformation (15) is amplified by the amplitude of the element marked on the average of the amplitudes.

Mathematically:

$$|\psi_t\rangle = \sum_{x=0}^{N-1} \alpha_x|x^{"}\rangle \quad ,$$
$$U_s = (2(\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle)(\frac{1}{\sqrt{N}}\sum_{x'=0}^{N-1}\langle x'|) - 1) \tag{16}$$

therefore

$$U_s|\psi_t\rangle = (2(\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle)(\frac{1}{\sqrt{N}}\sum_{x'=0}^{N-1}\langle x'|) - 1)\sum_{x=0}^{N-1}\alpha_x|x^{"}\rangle$$

or

$$U_s|\psi_t\rangle = (\frac{2}{N}\sum_{x=0,x\prime=0,x\,=0}^{N-1}\alpha_x|x\rangle\langle x'|x^{"}\rangle - \sum_{x=0}^{N-1}\alpha_x|x\rangle$$

or

$$U_s|\psi_t\rangle = (\frac{2}{N}\sum_{x=0}^{N-1}\alpha_x^{"}\sum_{x=0}^{N-1}|x\rangle - \sum_{x=0}^{N-1}\alpha_x|x\rangle)$$

from where

$$U_s|\psi_t\rangle = \sum_{x=0}^{N-1}(2\sum_{x''=0}^{N-1}\frac{\alpha_x^{"}}{N} - \alpha_x)|x\rangle = \sum_{x=0}^{N-1}(2A - \alpha_x)|x\rangle$$

Where A is the average of each α_x .

$$A = \sum_{x''=0}^{N-1}\frac{\alpha_x^{"}}{N} \tag{17}$$

After some reductions by the summations and the average for the state $|\psi_{t+1}\rangle$ we have:

$$|\psi_{t+1}\rangle = U_s U_f|\psi_t\rangle = \begin{cases} \frac{2^{n+1}+2^n-4}{2^n\sqrt{2^n}}|x\rangle \text{ if } x = w \\ \frac{2^{n+1}-2^n-4}{2^n\sqrt{2^n}}|x\rangle \text{ if } x \neq w \end{cases} \tag{18}$$

When implementing the algorithm in a quantum computer it repeats the three steps, and a response with a greater amplitude of probability is obtained.

4. Simulation of Grover's algorithm.

Grover's algorithm is a search algorithm that returns the match with the highest probability to be correct from an unordered sequence of data with N elements. The running time from the classical point of view has an order of 2^n but the quantum version has an order of $2n$. This simulation intends to show how the running time of a quantum algorithm in a classical computer is crucial when the goal is reducing response times.

This simulation consists of three important parts:

1. The initialization of the amplitudes of the different states obtained by the superposition, in other words, when the state of the system is for $t = 0$, $|\psi_{t=0}\rangle = |\psi_0\rangle = |s\rangle = h^{\otimes n}|0\rangle$
2. The reflection to all states, this means $|\psi_t\rangle = U_f|\psi_0\rangle$ where only the marked state will be affected.
3. And finally, the reflection that modifies all the states $|\psi_{t+1}\rangle = U_s|\psi_t\rangle$ completing this way the transformation.

4.1. Input of data

This simulation needs the number of Qbits that will consider executing the search. This will determine the number of records.

```
System.out.println("Enter number of
qubits:");

numQubits = scan.nextInt();

At the same time, it requires the
expected record:

System.out.println("Enter the value you.re
searching for:");

value = scan.nextInt();
```

4.2. Data processing

This process starts by executing the initializeBinary() method which will allow printing the bit string provided by the user in binary, getting as a result a bra-ket notation array using Walsh-Hadamar transform. (12)

```
public void initializeBinary()

\{

Integer i = 0;

double amplitude = 1 / Math.sqrt(Math.pow
(2, numQubits));

System.out.print(amplitude);

System.out.print("(");

for (i = 0; i \TEXTsymbol{<} Math.pow(2,
numQubits) - 1; i++)

\{

System.out.print("j" + Integer.
toBinaryString(i) + "\TEXTsymbol{>} + ");

\}

System.out.println("j" + Integer.
toBinaryString(i) + "\TEXTsymbol{>})");

\}

In addition, the simulation executes the
initialize() method with the purpose of
printing to the screen the bit string
in decimal numbers.

public void initialize()

\{

int i = 0;

double amplitude = 1 / Math.sqrt(Math.pow
(2, n)umQubits));

System.out.print(amplitude);

System.out.print("(");

for (i = 0; i \TEXTsymbol{<} Math.pow
(2, numQubits) - 1; i++)

\{

System.out.print("j" + i +
"\TEXTsymbol{>} + ");

\}

System.out.println("j" + i +
"\TEXTsymbol{>})");

\}

After that, the phaseInversion() method is
called and this method will be responsible
for the phase inversion equation (\ref{14x})
from Grover's algorithm, obtaining as an
output, the same array but with the sign
value changed.

public void phaseInversion()

\{

double element = 0;

element = array.get(value);

System.out.println(element+"element ok");

if (element \TEXTsymbol{<} 0);

else

\{ \qquad

element = -1 * element;

array.set(value, element);

System.out.println("array else"+array);

\}

\}
```

At the same time, the inversionMean() method is being called which calculates the reflection over the mean value, equation (16). The output shows the change of the amplitudes from the initial array, so the chance that the state collapse is higher.

```
public void inversionMean()

\{

double average = 0;

double element = 0;

for (int i = 0; i \TEXTsymbol{<}
array.size(); i++)

\{

average += array.get(i);

\}

average = average / array.size();

for (int j = 0; j \TEXTsymbol{<}
array.size(); j++)

\{

element = array.get(j);

element = (average - element)
+ average;

array.set(j, element);

System.out.println("array"+array);

\}

\}
```

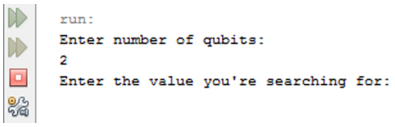
4.3. Output

Finally, the simulation calculates the probability of using the amplitude to the second power using the findProbability() method.

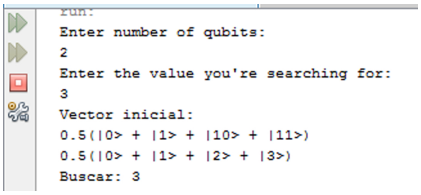
```
public double findProbability()

\{

double probability = vector.get(value)/
(Math.sqrt((Math.pow(2, numQubits))));

probability = Math.pow(probability, 2);

return probability;

\}

And from the createArray() method, it is
created a matrix of 2n slots obtaining the
new array that will be sent to Grover's
algorithm

public ArrayList\TEXTsymbol{<}
Double\TEXTsymbol{>} createArray()

\{

ArrayList\TEXTsymbol{<}Double\TEXTsymbol{>}
newVector = new ArrayList%
\TEXTsymbol{<}Double\TEXTsymbol{>}();

for (int i = 0; i \TEXTsymbol{<} Math.pow
(2, numQubits); i++)

\{

newVector.add(1.0);

\}

return newVector;

\}
```

Example

Step 1. Capture the size of the vector and the number you want to find:

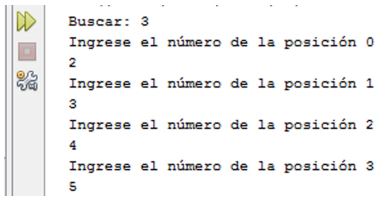


Step 2. View of the initial vector in decimal and in binary:

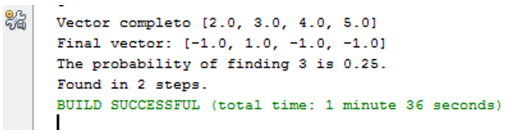


 DOI: http://dx.doi.org/10.1590/1806-9126-RBEF-2019-0115

Step 3. Capture of the numbers that the vector will store:



Step 4. View of the complete vector and the final vector after the completion of the search:



4.4. Comparison of results with other classical algorithms

The simulation of a quantum search using Grover's algorithm (G) and the simulations of sequential search (SS) and binary search (BS), allow determining the efficiency of the quantum algorithm compared to a simple search algorithm. After analyzing five different sizes of arrays, each with a sample of 20 measurements, it is possible to obtain an average which is illustrated in the next table. In the three algorithms, the simulation uses an unordered list with N elements.

vector size	algorithm	# of interactions	response time
4	G	2	0,0028369
	BS	4	0,3298
	BB	3	0,28
16	G	4	0,00328628
	BS	16	0,3845
	BB	5	0,3
32	G	5	0,003958214
	BS	32	0,3911
	BB	6	0,3096
64	G	8	0,00523802
	BS	64	0,3997
	BB	10	0,3141
128	G	11	0,01546982
	BS	128	0,4009
	BB	12	0,3299

5. Conclusions

To conclude, the development of this simulation allowed understanding the scope and the potential that quantum computing may have in future's technology, it is now only necessary to create new tools that will implement algorithms such as Grover's, which not only seek to speed up and improve response times when solving mathematical calculations, but also, to start being a fundamental part of new software development that can solve many problems as computer security , where these new algorithms will be the first choice and eventually leaving behind classical computing.

Once again, it is important to point out that the potential and the speed of the Grover Algorithm and other quantum algorithms [9] lies in the fact that the principle of quantum superposition is exploited to the maximum exploiting the possibility of a coherent quantum superposition of states as input. Therefore, the quantum algorithm allows solving the problem in exponentially less time if compared to the classical procedure.

The authors thank the reviewer for the observations.

References

[1] P. Benioff. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by
Turing machines. J. Stat. Phys. 22, 563–591 (1980).
[2] P. Benioff. Quantum mechanical Hamiltonian models of Turing machines. J. Stat. Phys. 29, 515–546 (1982).
[3] P. Benioff. Quantum mechanical models of Turing machines that dissipate no energy. Phys. Rev. Lett. 48, 1581–1585 (1982).
[4] R. P. Feynman. Simulating physics with computers. Int. J. Theor. Phys. 21, 467–488 (1982).
[5] R. P. Feynman. Quantum Mechanical Computer. Foundations of Physics ,1986, Volume 16, Issue 6, pp 507–531
[6] V. Moret-Bonillo. Principios fundamentales de computación cuántica.(2013).
[7] Zhuang Jiaya; et al, Analysis and Simulation of Grovers algorithm, International Journal of Machine Learning and Computer,vol 4 No1 2014
[8] A.B Mutibara and R.Refianti. Simulation of Grovers algorithm Quantum search in a Classical Computer, International Journal of computer Scince and Information security, Vol 8 No9, 2010.
[9] Yohan Vianna , Mariana R. Barros, Malena Hor-Meyll. Classical realization of the quantum Deutsch algorithm, American Journal of Physics, 86, 914 (2018);
[10] Wang, D.S., Hill, C.D and Hollenberg, L.C.L. Simulations of Shor's algorithm using matrix product states, Quantum Inf Process (2017) 16: 176
[11] Johansson, N. & Larsson, JÅ, Efficient classical simulation of the Deutsch–Jozsa and Simon's algorithms, Quantum Inf Process (2017) 16: 233
[12] Roger Penrose, García Sanz Javier. Las sombras de la mente. Crítica, (1996).
[13] H.T. Schmid et al.,.Evidence of wave-particule duality for single fast hidrogen atom. Phys. Rev. Lett. 101.083201(2008)