Performance of IBM's open-access quantum computers on different quantum circuits

Author: Alex Martínez Miguel.
Facultat de Física, Universitat de Barcelona, Diagonal 645, 08028 Barcelona, Spain.[*]

Advisor: Artur Polls Martí

(Dated: January 16, 2020)

Abstract: We use the different quantum computers which IBM made available on the cloud (IBM Q Experience) to implement different quantum circuits. First, we study the dynamics of a single spin interacting with a magnetic field using the main single-qubit gates, which allows us to obtain the evolution of the three components of the spin. Finally, we study the performance of the quantum computers by implementing two quantum algorithms: the dense coding protocol, which we use to study the connectivity between qubits, and the quantum Fourier transform, which we use to study the performance of three different quantum computers when the number of qubits is increased.

I. INTRODUCTION

In the last four decades the field of quantum computation has not stop growing and attracting people from different backgrounds, such as physics, computer science, information theory and cryptography [1]. Nowadays, it is a very promising and exciting field. Quantum computers have been presented as potential tools for solving real problems which are considered out of reach for classical computers. It is believed that they could contribute to the development of new breakthroughs in science, medications, machine learning methods or materials to make more efficient devices and structures, among others. One such example would be a quantum algorithm for factoring integers [2], which was developed by Peter Shor in 1994.

To that end, big companies like IBM or Google [3] have been investing huge amounts of money to build the first full-operational quantum computer. There is more, in May 2016, IBM launched the IBM Q Experience [4], an open online platform which gives the general public access to a set of prototype quantum processors.

The main purpose of this work is to implement well-known quantum algorithms in the framework set up by IBM. On the one hand, a study regarding the interaction of a single spin with a magnetic field will be done. On the other hand, an inspection of the performance of these processors will be carried out by going through two specific quantum algorithms: superdense coding and quantum Fourier transform. From the set of quantum processors we will use three of them: two 5-qubit prototypes, ibmqx2 and ibmq_vigo, and the 15-qubit prototype ibm_16_melbourne.

The study is organised as follows. In Section II, the concept of qubit will be explained and a brief description of the most relevant (to this work) quantum gates will be made. In Section III, we study the interaction of a spin with a magnetic field. In Section IV, we use two quantum algorithms to analyze the performance of different quantum prototypes. Finally, in Section V, conclusions are presented.

II. FUNDAMENTALS

A. The qubit

Quantum computers use qubits, which generalize the classical bit. They are two-level systems, i.e. systems with one quantum property that can take two possible values. These two values allow us to label the basis states and are denoted as $|0\rangle$ and $|1\rangle$. The main feature of the qubits is the possibility to have superposition states. A qubit can be in an arbitrary combination of the two basis states

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

being α and β complex numbers. Taking into account the normalization condition $\langle\psi|\psi\rangle = |\alpha|^2 + |\beta|^2 = 1$, it follows that a general state of a qubit can be expressed in terms of two real parameters (up to a global phase)

$$|\psi\rangle = \cos(\theta/2)|0\rangle + \sin(\theta/2)e^{i\phi}|1\rangle. \tag{2}$$

Therefore, the quantum state of a single qubit can be visualized as a vector of length 1 inside the so-called Bloch sphere. The parameters θ and ϕ are the polar and azimuthal angle of the spherical coordinates.

In the IBM Q Experience, the prototypes use a physical type of qubit called superconducting transmon qubits, which are made of superconducting materials. The two states $|0\rangle$ and $|1\rangle$ represent two possible energy levels within this superconducting system. For these systems to behave as an abstract qubit they must be at drastically low temperatures (15 mK in the case of IBM's).

B. Quantum gates

Quantum gates are unitary matrices (they must be reversible and conserve the probability amplitudes) which

[*] Electronic address: `amartimi25@alumnes.ub.edu`

allow to change the state of the qubit, e.g. create superpositions, rotations, etc.
The most basic quantum gates are the single-qubit gates X,Y and Z (the usual Pauli matrices). In matrix form

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{3}$$

Two other important gates are the Hadamard gate, H, which generates superposition, and the S gate, which adds a phase. In matrix form

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \tag{4}$$

The Hadamard gate acts as follows: $H|0\rangle = |+\rangle \equiv (|0\rangle + |1\rangle)/\sqrt{2}$ and $H|1\rangle = |-\rangle \equiv (|0\rangle - |1\rangle)/\sqrt{2}$. Applying the the S gate to these new states: $S|+\rangle = |\circlearrowright\rangle \equiv (|0\rangle + i|1\rangle)/\sqrt{2}$ and $S|-\rangle = |\circlearrowleft\rangle \equiv (|0\rangle - i|1\rangle)/\sqrt{2}$. We are able to define three different bases: the computational basis: $\{|0\rangle, |1\rangle\}$, the x-basis: $\{|+\rangle, |-\rangle\}$ and the y-basis: $\{|\circlearrowright\rangle, |\circlearrowleft\rangle\}$. In Section III we will see how to make measurements in each one of the bases.
To end with the single-qubit gates, we must introduce the U_1, U_2 and U_3. In matrix form

$$U_3(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\theta/2) & -e^{i\lambda}\sin(\theta/2) \\ e^{i\phi}\sin(\theta/2) & e^{i\lambda + i\phi}\cos(\theta/2) \end{pmatrix}, \tag{5}$$

$$U_1(\lambda) = U_3(0, 0, \lambda), \quad U_2(\phi, \lambda) = U_3(\pi/2, \phi, \lambda). \tag{6}$$

All the aforementioned gates are just special cases of these.

Now lets turn to the the two-qubits quantum gates. The most important one is the Controlled-NOT (CNOT) gate. Its action is to flip the target qubit if the control qubit is $|1\rangle$; otherwise it does nothing. The last quantum gate useful for this work is the controlled phase shift gate R_m, which is specially relevant for implementing the quantum Fourier transform. In matrix form they read

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad R_m = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{\frac{2\pi i}{2^m}} \end{pmatrix}. \tag{7}$$

III. SINGLE-SPIN DYNAMICS

Firstly, we study the interaction of a single spin with a time independent magnetic field along the z-direction $\vec{B} = B_0\hat{z}$. The objective is to analyze how to build the quantum circuits that allow us to compute the three components of the spin as a function of time.
Given the magnetic moment of the spin $\vec{\mu} = g\mu_B\vec{S}/\hbar$, which is expressed in terms of the Pauli matrices σ_i from $S_i = \sigma_i\hbar/2$, the Hamiltonian describing this interaction is

$$\mathcal{H} = -\vec{\mu} \cdot \vec{B} = \frac{\hbar\omega_0}{2}\sigma_z, \tag{8}$$

where $\omega_0 = -g\mu_B B_0/\hbar$ is the Larmor frequency. Since we are dealing with a time-independent Hamiltonian, the solution of the Schrödinger equation for this system might be found applying the time evolution operator [5] $U(t, 0) = \exp(-i\mathcal{H}t/\hbar)$ to the initial state $|\psi(0)\rangle$

$$|\psi(t)\rangle = \exp(-i\mathcal{H}t/\hbar)|\psi(0)\rangle. \tag{9}$$

In matrix form, the operator is written as

$$U(t) = \exp\left(-i\frac{\omega_0 t}{2}\sigma_z\right) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\omega_0 t} \end{pmatrix}. \tag{10}$$

We choose our initial state to be in a superposition of both basis states

$$|\psi(0)\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + \sin\left(\frac{\theta}{2}\right)|1\rangle. \tag{11}$$

Thus, the evolved state will be given by

$$|\psi(t)\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\omega_0 t}\sin\left(\frac{\theta}{2}\right)|1\rangle. \tag{12}$$

Now, by computing the different components of the spin we get

$$M_z(t) = \langle\psi(t)|S_z|\psi(t)\rangle = \frac{\hbar}{2}\cos(\theta), \tag{13}$$

$$M_x(t) = \langle\psi(t)|S_x|\psi(t)\rangle = \frac{\hbar}{2}\sin(\theta)\cos(\omega_0 t), \tag{14}$$

$$M_y(t) = \langle\psi(t)|S_y|\psi(t)\rangle = \frac{\hbar}{2}\sin(\theta)\sin(\omega_0 t). \tag{15}$$

Next, we are ready to implement the problem in a quantum circuit. First, we need to prepare the initial state. All circuits in the IBM Q Experience are initialized in the $|0\rangle$ state, so, from this state we can obtain $|\psi(0)\rangle$ by applying the following unitary operation

$$U_{\text{in}} = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}, \tag{16}$$

which is obtained by using the $U_3(\theta, \varphi, \lambda)$ quantum gate and setting $\varphi = 0$ and $\lambda = 0$. The next step is to implement the time evolution operator, which is given by Eq. (10). This is accomplished by using the $U_1(\lambda)$ if we identify $\lambda = \omega_0 t$. Finally, we must make a measurement. IBM Q Experience only allows to make a z-measurement directly, which is the one we must use to compute $M_z(t)$. To find the x-component of the spin, we must measure in the x-basis, and this is done using circuit (b) of Fig. 1. Same for the y-component, the measurement in the y-basis is shown in circuit (c) of Fig. 1.

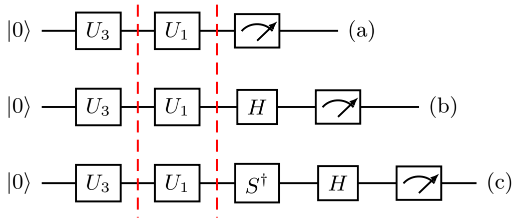

FIG. 1: Circuits used for measuring the components of the spin: (a) is a z-measurement, (b) is a x-measurement and (c) is a y-measurement. The red dashed line divides the circuit in three parts: preparation of the initial state, implementation of the time evolution operator and measurement.

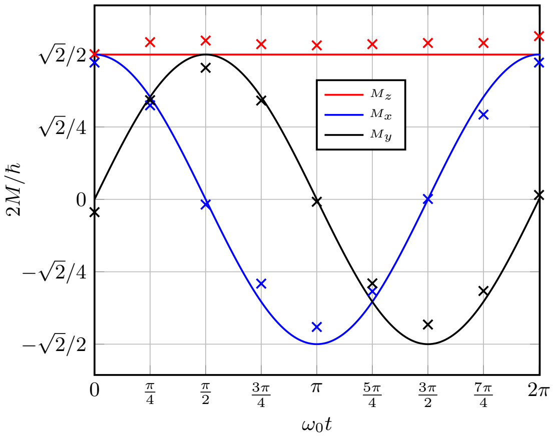

FIG. 2: Expected value of the three components of the spin as a function of $\omega_0 t$ using the prototype ibmqx2 after 8192 shots each point. Continuum lines are the theoretical results, and points are experimental data. In this case $\theta = \pi/4$.

The experimental points reflect rather well the theoretical behavior in all three cases.
Note that we have not simulated the time evolution of the system using the quantum computer. Instead, we have constructed the evolved state with a quantum circuit and found the probability of having the system at each one of the basis states at a given time, which we introduce manually.

IV. PERFORMANCE OF THE QUANTUM COMPUTERS

A. Superdense coding

Superdense coding is a protocol introduced by Bennett and Wiesner [6] which allows to send two bits of classical information between two partners (Alice and Bob) who share an Einstein–Podolsky–Rosen (EPR) state by only manipulating and sending one qubit. The protocol works as follows [7] (see Fig. 3):

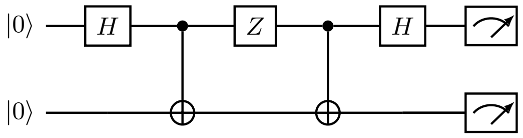

FIG. 3: Quantum circuit for the implementation of dense coding. In this case, the string of classical bits 10 is sent.

1. Alice and Bob share an EPR state, which is prepared, for instance, in the state

$$|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{17}$$

This is accomplished through the Hadamard and CNOT gate. It is important to notice that the preparation of the EPR pair might be done by a third party, who will send one qubit to Alice and the other to Bob. Thus, they can be located far apart.
2. Depending on the values of the two classical bits (00, 01, 10 and 11) that Alice wants to send to Bob, she will perform different unitary operations on her qubit. If she wishes to send 00, she does not need to do anything, i.e. the identity matrix. In case it is the 01 bit string, she applies the Pauli matrix X. Another possibility would be the 10, in which case Alice would have to operate with the Pauli matrix Z. Finally, to send the 11, the gate iY is needed, which is accomplished by the product $Z \cdot X$.
3. Alice sends her qubit to Bob.
4. Finally, Bob can determine which one of the four possible bit strings Alice sent by using a CNOT and Hadamard gate again and making a measurement.
In conclusion, Alice is able to send a two-bit string of information by only interacting with one qubit. Note that two qubits are involved in the protocol, but Alice never needs to interact with one of them.

Our purpose is to analyze this quantum algorithm using different pairs of qubits. As it can be seen in Fig. 4, not all qubits have an allowed connectivity between them due to experimental constraints, so we expect different results if we use two qubits with an allowed connectivity, e.g. 1 and 2, or two qubits with a not-allowed connectivity, e.g. 1 and 3.
Since the theoretical outcome of receiving the same bit string that was sent is 100%, we can take the probability outcomes of Table I as a measurement of the performance of the quantum computer, which varies depending of the pair of qubits used.

As expected, the performance is better when we use qubits with an allowed connectivity. Let us emphasize that two not-connected qubits can actually be used to perform two-qubit operations but, as we have seen,

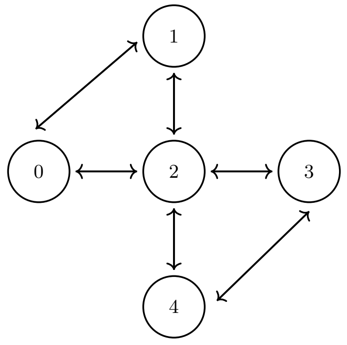

FIG. 4: Connectivity of the qubits in the ibmqx2 quantum prototype

Bits sent	Allowed connectivity	Not-allowed connectivity
00	$96,8 \pm 0,3\%$	$93 \pm 2\%$
01	$72,9 \pm 0,6\%$	$71,2 \pm 0,9\%$
10	$94,8 \pm 0,2\%$	$91,4 \pm 0,9\%$
11	$93,6 \pm 0,4\%$	$88 \pm 3\%$

TABLE I: Mean probability outcomes of receiving the same two classical bits which were sent after 5 runs of 8192 shots using the protoype ibmqx4. The errors are the standard deviations among these five runs. The second column shows the results using qubits 1 and 2 (allowed connectivity) and the third column show the results using qubits 1 and 3 (not-allowed connectivity).

the performance degrades in this case.
It might be surprising that the performance decreased significantly when we sent the classical bits 01, which corresponds to the case where we used the X gate. A possible explanation to this could be that the implementation of this gate is worse than the others.

B. Quantum Fourier Transform

To introduce the quantum Fourier transform (QFT) [8], first we need to define the discrete Fourier transform, which takes a vector of N complex numbers, $(a_0, \ldots, a_{N-1})$, and returns another vector of complex numbers, $(b_0, \ldots, b_{N-1})$, according to

$$b_j \equiv \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} a_k e^{2\pi i k j/N}, \tag{18}$$

where i is the imaginary unit. The QFT acts on a quantum state $|\psi\rangle = \sum_{j=0}^{N-1} a_j |j\rangle$ and maps it to another quantum state $\mathcal{F}|\psi\rangle = |\phi\rangle = \sum_{j=0}^{N-1} b_j |j\rangle$ where b_j is given by Eq. (18). In our case, N is the number of basis states, which are $N = 2^n$ with n being the number of qubits. The basis states $|j\rangle = |j_1 j_2 \ldots j_n\rangle$ enumerate all possible states of the system, where it is useful to use the following binary to decimal representation $j = \sum_{i=1}^{n} j_i 2^{n-i}$.
Let us try to find the matrix form of the algorithm. If we act on a basis state $|k\rangle$, we get

$$\mathcal{F}|k\rangle = \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} e^{2\pi i jk/2^n} |j\rangle. \tag{19}$$

We can now find the different matrix elements by computing $\langle k|\mathcal{F}|0\rangle$, $\langle k|\mathcal{F}|1\rangle$ and so on until $\langle k|\mathcal{F}|2^n-1\rangle$ with k taking values from 0 to $2^n - 1$. We find

$$\mathcal{F} = \frac{1}{2^{n/2}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{2^n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(2^n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{2^n-1} & \omega^{2(2^n-1)} & \cdots & \omega^{(2^n-1)(2^n-1)} \end{pmatrix}, \tag{20}$$

where we have defined $\omega \equiv e^{2\pi i/2^n}$. From this it is easy to see that it is a unitary transformation.

Regarding the circuit implementation, it is easier to identify the circuit if we study the effect of the QFT on one of the basis states and then let the mixed states be defined by linearity. Following the procedure described in Ref. [8], we find

$$\mathcal{F}|k_1 k_2 \ldots k_n\rangle = \frac{1}{2^{n/2}}(|0\rangle + e^{2\pi i[0.k_n]}|1\rangle) \otimes(|0\rangle + e^{2\pi i[0.k_{n-1}k_n]}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i[0.k_1 k_2 \ldots k_n]}|1\rangle), \tag{21}$$

where we have introduced a new compact notation $[0.k_1 \ldots k_n] = \sum_{l=1}^{n} k_l 2^{-l}$.
The product representation of Eq. (21) makes it easier to build the quantum circuit that computes the QFT. Such circuit is shown in Fig. 5, where we have introduced the R_m operator given by Eq. (7).
The fact that this algorithm can be implemented using an arbitrary number of qubits might be used to inspect the effect of increasing the number of qubits on the performance of the algorithm. To this purpose, we create an initial state which is a superposition of all the basis states

$$|\psi_0\rangle = \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} |k\rangle. \tag{22}$$

This is done by applying Hadamard gates to all the qubits. Then, the action of the QFT on $|\psi_0\rangle$ is always the same: $\mathcal{F}|\psi_0\rangle = |0\rangle$. We can take the same argument as in Section IV A and use the probability outcomes as a measurement of the performance.
We implement the algorithm using from 1 to 5 qubits with three different quantum computers, each one having a different connectivity.

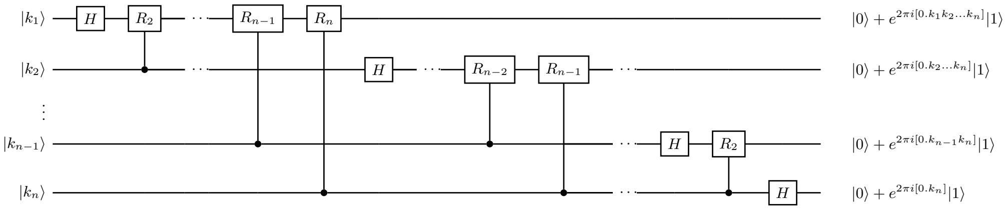
$|k_1\rangle$
$|k_2\rangle$
$\vdots$
$|k_{n-1}\rangle$
$|k_n\rangle$
H
R_2
R_{n-1}
R_n
H
R_{n-2}
R_{n-1}
H
R_2
H
$|0\rangle + e^{2\pi i[0.k_1k_2\ldots k_n]}|1\rangle$
$|0\rangle + e^{2\pi i[0.k_2\ldots k_n]}|1\rangle$
$|0\rangle + e^{2\pi i[0.k_{n-1}k_n]}|1\rangle$
$|0\rangle + e^{2\pi i[0.k_n]}|1\rangle$

FIG. 5: Circuit for the implementation of the quantum Fourier transform. To obtain the state in Eq. (21) from this circuit, swap operations of the qubits must be performed to reverse their order.

The performance decreases as the number of qubits increases, as seen in Fig. 6. This should not be surprising, since with the number of qubits it also increases the number of quantum gates, which means more manipulation of the qubits.
We find a similar behavior for two 5-qubit quantum computers, ibmqx2 and ibmq_vigo. The performance is reasonable, i.e. error below 20%, for three or less qubits, and degrades for four and five qubits with errors above 80%. A different behavior is observed for the larger machine, the 15-qubit ibm_16_melbourne. In this case, the error grows linearly with the number of qubits.

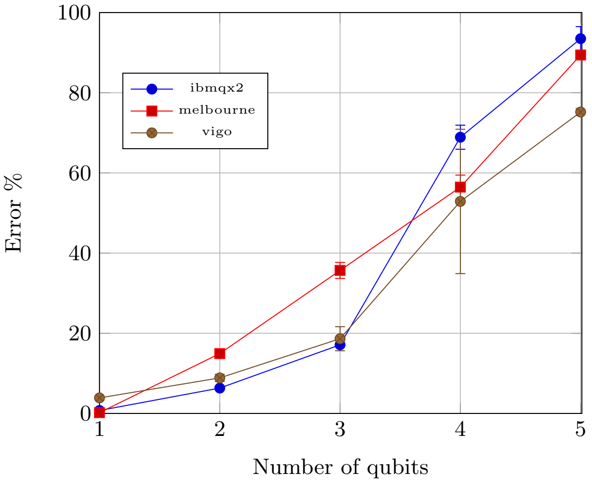

FIG. 6: Difference between the theoretical outcome and the mean probability outcome as a function of the number of qubits after 5 runs of 8192 shots for each qubit and for each quantum computer. The error bars are the standard deviations among these five runs.

V. CONCLUSIONS

In this work we have benchmarked the state of the art IBM quantum computers available online implementing several relevant quantum algorithms. First, we have explored the performance of a single qubit. To this aim, we have considered the dynamics of a spin interacting with a magnetic field. The obtained performance of the quantum computers in this case is very satisfactory, closely following the expected behavior. Then we have concentrated on two well-known multiqubit algorithms, superdense coding and the quantum Fourier transform. The first has been used to explore the importance of the physical connectivity of the qubits for the performance of the protocols. Indeed, physical connectivity results in a better performance. The Fourier transform has been implemented on three different quantum computers. The performance is found to be reasonable for three or less qubits in the five qubit computers.

Acknowledgments

I would like to thank Artur Polls and Bruno Juliá for their guidance and advice during the development of this work. I would also like to express my gratitude to my family and friends, who have always been of great support.

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, (Cambridge University Press, Cambridge 2010, 10th Anniversary Edition).
[2] P.W. Shor, SIAM J. Sci. Statist. Comput. **26**, 1484 (1997).
[3] F. Arute, K. Arya, R. Babbush, et al. Nature **574**, 505 (2019).
[4] IBM. *IBM Quantum Experience.* Retrieved from https://quantum-computing.ibm.com (2016).
[5] J.J. Sakurai and J. Napolitano, *Modern Quantum Mechanics*, (Addison-Wesley, San Francisco 2011, 2nd. ed.).
[6] C. Bennett and S. Wiesner, Phys. Rev. Lett. **69**, 2881 (1992).
[7] G. Benenti, G. Casati and G. Strini. *Principles of Quantum Computation and Information*, (World Scientific Publishing Co. Pte. Ltd., Singapore 2004, 1st. ed.).
[8] D. Bacon, *The Quantum Fourier Transform.* Department of Computer Science & Engineering, University of Washington (2006).