Randomness in quantum computation

Juan Pablo Paz[1,*]

[1] *Theoretical Division, LANL, MSB213, Los Alamos, NM 87545, USA*

(Dated: November 13, 2003)

PACS numbers:

In ordinary computers, it is sometimes useful to flip a coin. Non–deterministic algorithms (those programs that involve making such random choices at some steps) can be far more efficient in finding solutions to a variety of problems [1]. The generalization of this kind of methods seems to be useful also for quantum computers where some randomized algorithms have recently been proposed [2, 3]. However, the situation in quantum computation is more involved: To introduce a random evolution in a quantum computer one needs much more than a coin, a dice or a roulette! In fact, the operations available for a quantum computer are not only infinite but also form a continuous set (i.e., they are uncountably infinite). However, the recent work by J. Emerson and coauthors [4] shows that enforcing a (pseudo) random evolution on a quantum computer is not as hard as one may believe. In fact, in their paper the authors present a simple method, an algorithm, to enforce a quasi random evolution by only acting on individual quantum bits and controlling simple two–body interactions between neighboring qubits.

The usefulness of coin–flipping in classical computation could be rather counterintuitive at first sight. How is it, one may ask, that we can resort to randomness to find the answer to a well defined mathematical problem? Randomized algorithms turn out to be efficient ways of finding solutions to some problems if we can tolerate erroneous answers with low probability. This is indeed the case for some problems for which it is easy to check if the solution provided by our computer, which may be erroneous, is indeed a correct solution. (Some problems of this class, which is known as NP [1], can be more efficiently solved by randomized algorithms than by deterministic ones.) Coin–flipping is also useful in classical computation when performing calculations that involve statistical sampling over many realizations of some process. This is what one is typically interested in doing when using computational tools to study properties of complex natural systems. In this context, the most popular coin–flipping tool is the technique named after the most famous roulette: the Monte Carlo method.

Quantum computers are believed to be much more efficient than classical ones [5]. Randomness enters naturally in quantum computation as it is inherent to quantum systems, which typically give different answers when subject twice to the same measurement. Due to this fact, in a typical quantum algorithm the result of the final measurement of the state of the computer is not always the same. Different results are obtained with a probability distribution that encodes the answer to the problem at hand. Quantum algorithms are cleverly designed in such a way that by running the computation a number of times, the final measurement reveals just enough properties of the probability distribution to answer the desired question (and, for example, factor an integer into prime numbers). But a typical quantum algorithm is not at all random, since at every stage before the final measurement the quantum computer evolves according to well defined deterministic rules. The evolution of the state of the quantum computer is described by a unitary operator which, for a system with n quantum bits, is represented by an $N \times N$ unitary matrix (a member of the group $U(N)$, where $N = 2^n$). Thus, there would be nothing random in the evolution of a quantum computer running a typical quantum algorithms.

However, the idea of using random unitary operators in some quantum algorithms has been recently proposed. Some solid results have been established in the area of quantum communication. Indeed, Charles Bennett, Peter Shor and their coworkers [2, 3] showed that using random operations one can decrease the communication cost to achieve tasks such as remote state preparation or to construct more efficient quantum data hiding schemes. Also, it is believed that the use of random unitary operators may be essential to achieve more efficient ways of characterizing the way in which the quantum computer is affected by the interaction with its environment. In general, a complete characterization of the decoherence process[6] induced by the environment would require the costly application of both process and state tomography [5, 7]. However, as Emerson et al argue [4] averaging over random unitaries, simple benchmarking tools such as fidelity or purity decay could become independent of the target operation and properly characterize the most important aspects of decoherence.

But how is it that one can induce a quantum computer to evolve in a random way? As mentioned above, this is much harder than flipping a coin. The reason is that all $N \times N$ unitary matrices describe quantum evolutions that are in principle allowed. To sample this continuous set seems to be hard, requiring a number of elementary operations which was believed to be huge (close to $N^2 \log^2(N2)$). The results reported by Emerson et al [4] show that there is a remarkably simple alternative to this. In their paper, these authors show that it is pos-

*Electronic address: jpaz@lanl.gov

sible to device a simple set of operations that generate an ensemble of pseudo–random unitaries with the same coarse statistical features of the uniform ensemble over the group $U(N)$. To achieve this, it is necessary to perform m iterations of a proceedure that consists of two steps. The first step is to rotate randomly each quantum bit, which can be done by applying a random magnetic field or a sequence of laser pulses with random intensities (the direction and the angle of rotation of each qubit are $3n$ random variables which are the only inputs the method requires). The second step is to induce an interaction between neighboring pairs of qubits. This interaction creates entanglement between the quantum bits and is of a very simple nature (in technical terms, it can be described by the operator $U = \exp(i\pi \sum_j \sigma_z^j \otimes \sigma_z^{j+1}/4)$ where σ_z^j is the Pauli matrix of the j–th qubit). Remarkably, this is all one needs to get a fair approximation to a random ensemble: single qubit rotations and simple two–body interactions. In their paper, Emerson et al show that the method produce an ensemble with the properties one is mostly interested in for quantum information. For example, they show that the entanglement of the states obtained by applying the above process to simple initial states is indistinguishable from the one corresponding to a truly random ensemble (the convergence between the two ensembles is exponentially fast in m, the number of iterations). Also, compelling evidence is shown pointing towards the fact that the distribution of matrix elements of the unitary operator obtain by the above proceedure rapidly approaches that of a uniform ensemble. The simplicity of the method is ilustrated by the authors of [4] by actually implementing it in a toy quantum information processor using liquid state NMR techniques.

Quantum coin–tosses seem now efficiently implementable, at least in an approximate but still useful way. Their application in the design of new quantum algorithms and in designing new benchmarking techniques for characterization of decoherence will certainly be an active area of research in the near future.

[1] C.H. Papadimitrou, *Computational Complexity*, Addison Wesley (Reading, Mass, 1994).

[2] C.H. Bennett, P. Hayden, D. Leung, P. Shor and A. Winter, quant-ph/0307100.

[3] P. Hayden, D. Leung, P. Shor and A. Winter, quant-ph/0307104.

[4] J. Emerson, Y. Weinstein, M. Saraceno, S. Lloyd and D. Cory, Science (2003).

[5] M. Nielsen and I. Chuang, *Quantum Information and Computation*, (Cambridge University Press, Cambridge, 2000).

[6] for a review see J. P. Paz and W. H. Zurek, in "Coherent matter waves, Les Houches Session LXXII", edited by R Kaiser, C Westbrook and F David, EDP Sciences, Springer Verlag (Berlin) (2001) 533-614.

[7] C. Miquel, J. P. Paz, M. Saraceno, E. Knill, R. Laflamme, C. Negrevergne, Nature (London) **418**, 59 (2002).