A few words about quantum computing, epistemology, repeatability and reproducibility

David R.C. HILL – Benjamin A. ANTUNES – Thomas CLUZEL – Claude MAZEL

Université Clermont Auvergne, CNRS, Clermont Auvergne INP,
Mines Saint-Etienne, LIMOS, 63000 Clermont-Ferrand, France
David.Hill@uca.fr*

1 Introduction

Epistemology is also known as Philosophy of Science. It is the part of philosophy where we study knowledge, its foundations, nature, scope and its limits. Methodology is a branch of epistemology where we study the research and analysis methods that are specific to a science or a discipline. We often see this term confusingly used instead of method (used to establish or demonstrate a truth, conducing our thoughts according to determined principles and steps applied in a specific order). Sometimes the 'logy' suffix is used to give a scientific gloss to terms where we should not… Karl Popper, a major Philosopher of Science of the 20th century, has mainly focused his work on the logic of scientific discovery [1]. He raised reproducibility to a major criterion for the scientificity of research studies. Since a decade, we observe a reproducibility crisis in many domains, computer science being one of those. The ACM terminology recently changed in 2020 to reflect this awareness of computer scientists and the evolution in the right direction to produce reliable results. Classical computer are deterministic machines, even when we run stochastic simulations. When pseudo-random numbers are properly used, we can precisely obtain bitwise identical results with a proper method and thus debug the scientific software being built, which is essential [2]. Quantum machines are stochastic by essence, each run will produce a potentially different result, but reproducibility (and not repeatability) remains the main criterion to check the quality of the quantum machines: do we obtain the same statistics the same scientific conclusion? After a short reminder of where quantum computing is coming from, we will evocate work in progress while simulating and testing of the Grover's algorithm on real quantum processors.

2 From quantum thoughts to prototypes

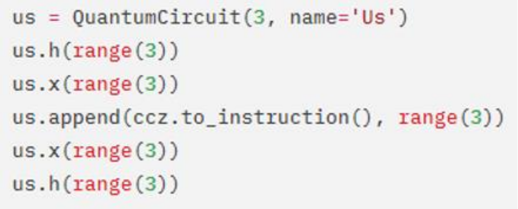
During a lecture at MIT in the beginning of the eighties, dealing with the possibilities and limits of numerical simulation [3], Richard Feynman raises a now famous question: can we simulate quantum systems in a probabilistic way with classical computers? We know that we do not scale and that many complexity barriers (exponential complexity in space and time…) exposed by Feynman led him to claim that we will not be able to imitate properly quantum mechanics. He opened the road towards the quest of quantum computing. Deutsch introduced three years later the notion of universal quantum Turing machine [4]. After the first theoretical contributions, concrete applications have pointed the tip of their nose 10 years later. Peter Shor's factoring algorithm contributed to dramatically accelerate the deciphering of several widely used encryption techniques. It is a strong reason why we have a race for universal quantum computing. Shor also contributed to Error correcting codes, which enabled to maintain the coherence of quantum computing (the latter being very unstable) [5]. This was an essential contribution of Shor less known than his proposal for a quantum factoring approach. At the same time, Lloyd showed how to simulate a quantum system with a mathematical object called a "Hamiltonian" [6], an operator giving the total energy of the system, and this approach enabled a better understanding of quantum systems. Even if we do not have at our disposal a universal quantum computer yet, we have interesting quantum circuits, allowing to implement algorithms according to different kinds of approaches depending from hardware vendors.

3 Reproducing the results of a small Grover's algorithm

As opposed to deterministic computers that propose repeatable results when executing computer programs, the quantum devices are truly stochastic; they come with a different result at each run executed in the same conditions. Results give statistical tendencies that we study statistically. For deterministic computers, "repeatability" (bitwise identical results) is essential, but it becomes tougher to obtain on huge parallel machines. For quantum devices and computers, which are stochastic, we only need reproducibility; meaning observing the same tendency and the same scientific conclusion is enough for corroboration.

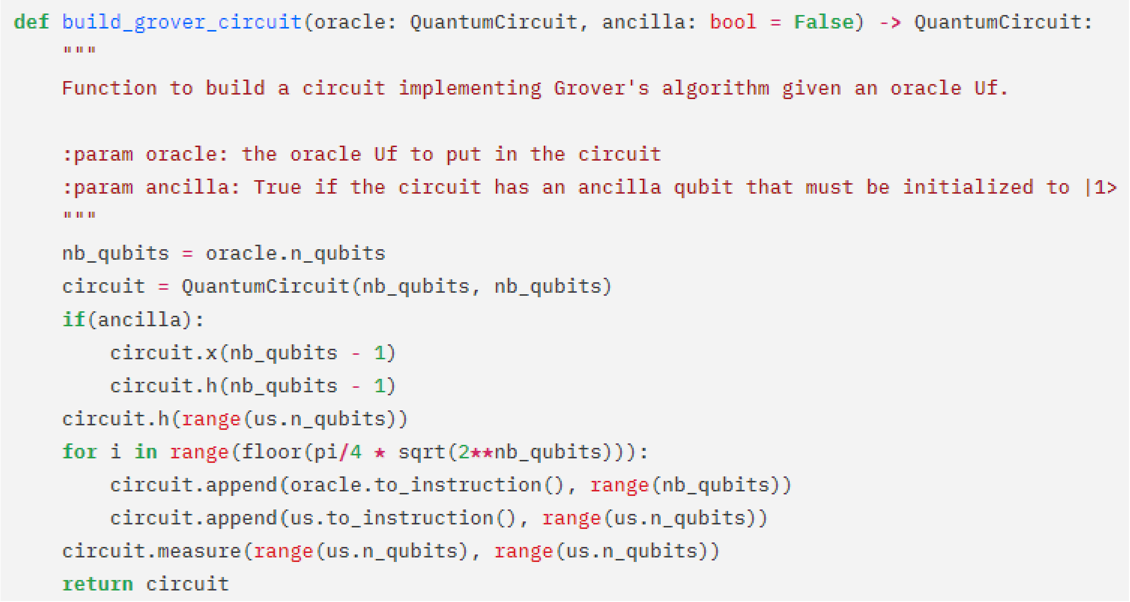
The definitions coined by the Association of Computing Machinery for repeatability, reproducibility and replicability changed in 2020. The reader interested in the terminology evolution can look at this keynote paper [7] where we present the main definitions around reproducibility. We have started to work on the reproducibility case of the Grover's algorithm already studied on 3 qubits by [8]. It is a search algorithm for unstructured data; its strength is that we obtain the result with a quadratic speedup in time complexity. After creating a superposition over all possible states in the database, the algorithm iterates and it amplifies the amplitude of the searched state which should "come out" with the highest probability before measurement.

For this small project, we have retained IBM quantum processors machines available freely through the cloud and the qiskit quantum computing software development framework also created by IBM. The latter also provides a quantum simulator. We have run the first tests in 2019 with satisfying simulations, but we were really disappointed by the results obtained when running on different real quantum machines [9]. Figure 1 gives the code of an operator (U_s) implemented with a CCZ gate in the IBM software environment and figure 2 presents an example of Python code to implement Grover’s algorithm. It can be parameterized to test different quantum gates. Three years after, we have started again our tests with a small number of qubits (3) since we rely on machines that are given for testing and training and a comparison with [8]. We want to see if we can find improvements on the machines available.



```
us = QuantumCircuit(3, name='Us')
us.h(range(3))
us.x(range(3))
us.append(ccz.to_instruction(), range(3))
us.x(range(3))
us.h(range(3))
```

Figure 1: Code corresponding to the operator used in the next function using a CCZ gate



```
def build_grover_circuit(oracle: QuantumCircuit, ancilla: bool = False) -> QuantumCircuit:
    """
    Function to build a circuit implementing Grover's algorithm given an oracle Uf.

    :param oracle: the oracle Uf to put in the circuit
    :param ancilla: True if the circuit has an ancilla qubit that must be initialized to |1>
    """
    nb_qubits = oracle.n_qubits
    circuit = QuantumCircuit(nb_qubits, nb_qubits)
    if(ancilla):
        circuit.x(nb_qubits - 1)
        circuit.h(nb_qubits - 1)
    circuit.h(range(us.n_qubits))
    for i in range(floor(pi/4 * sqrt(2**nb_qubits))):
        circuit.append(oracle.to_instruction(), range(nb_qubits))
        circuit.append(us.to_instruction(), range(us.n_qubits))
    circuit.measure(range(us.n_qubits), range(us.n_qubits))
    return circuit
```

Figure 2: Function generating the circuit of Grover's algorithm

IBM should produce quantum machines with 1000 qubits by the end of 2023 and Google announces at the beginning of this year that they made a significant contribution in correcting quantum errors. Encoding information onto multiple physical qubits to form a 'logical qubit' enables to correct errors. This approach is strongly considered to manufacture large-scale quantum computers with a level of error rate small enough for effective computing.

4 Conclusion

Three years after preliminary tests with the Grover's algorithm on IBM quantum processors, we are working on a small reproducibility project where we check if we have any progress on the quantum hardware proposed through cloud computing for training purposes. Our current testing shows the stability of simulation results, which confirm that our software implementation of the Grover's algorithm is correct. However, we are not able to obtain satisfactory results on real quantum machines for a small number of qubits and we even sometimes observed worse results than what we found in 2019, even when mitigating the errors. The technology freely at our disposal is probably the same. In our future work, we want to propose a method allowing another research team to test the reproducibility (and repeatability for the simulation side) of our results. This means working on an archive of our results with the corresponding code, the specification of the quantum machines, the versions of the qiskit simulator and the method applied to obtain what we observed.

References

[1] K. Popper (2005) : *The logic of scientific discovery*. Routledge, Basic Books Inc. New York.

[2] D.R.C. Hill (2015) : “Parallel random numbers, simulation, and reproducible research”, *IEEE CISE - Computing in Science and Engineering,* Invited Paper, **17(4),** 66–71.

[3] R. Feynmann (1982) : “Simulating Physics with computers, physics and Computation”, *International Journal of Theoretical Physics* **21**, 467–488.

[4] D. Deutsch (1985) : “Quantum Theory, the Church-Turing principle and the universal quantum computer”, *Proceedings of the Royal Society of London A,* **400(1818)**, 97–117.

[5] P.W. Shor (1996) : “Good quantum error-correcting codes exist”, *Physical review A,* **54(2)**, 1098–1106.

[6] S. Lloyd (1996) : “Universal quantum simulators”, *Science*, **273(5278)**, 1073–1078.

[7] D.R.C. Hill (2022) : “Reproducibility of simulations and High Performance Computing”, in *Proceedings of the European Simulation and Modeling,* October 26-28, 2022, ISEP, University of Porto, Porto, Portugal, 5–9.

[8] C. Figgatt, D. Maslov and K.A. Landsman et al. (2017) : “Complete 3-Qubit Grover search on a programmable quantum computer”. *Nature Communications* **8(1918)**.

[9] T. Cluzel, C. Mazel and D.R.C. Hill, : “Découverte de l'informatique quantique, état de l'art et tests sur machine IBM”, LIMOS Research Report – RR-2020-0304, 63 p.