Towards user-friendly quantum computing algorithms with the Qiwi library

QuantFi > qiwi

Quantum computing is *the* emergent computational power that we need to adress the challenges of the 21st century. As it is based on principles from *quantum mechanics*, it also carries with it a complete change in our way of designing new algorithms. Although for the moment quantum computers are still experimental and not ready for mass production yet, we have reached the point where *quantum algorithms* need to be developped forward, and integrated into a dedicated, hardware-agnostic, software framework.

IBM, a pioneer in quantum computing, has developped QISKIT, a set of software components that allow anyone to easily program IBM's quantum computers from anywhere in the world thanks to its cloud architecture. QISKIT components are concerned with the low-level intricacies of quantum circuits, transpilation (the quantum analogue of compilation), as well as quantum emulation (how to emulate the behavior of a quantum computer on classical hardware).

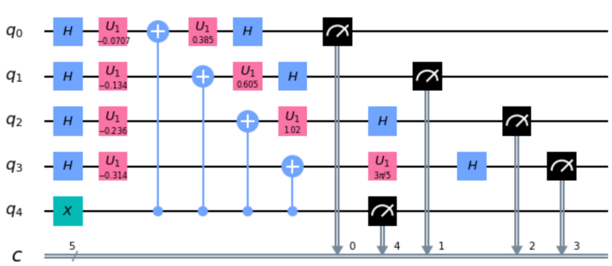

Figure 1: An example of a quantum circuit, rendered with the help of a QISKIT drawer.

Only the highest-level QISKIT module, named TERRA, proposes to its users a few basic quantum algorithms. However, at QuantFi we observed that all existing quantum algorithms can be constructed by applying meta-algorithms (such as amplitude amplification or phase estimation) on a few building blocks. In Qiskit (and similar libraries) these building blocks are not coded as gates in their optimal versions, and the meta-algorithms are just absent. We decided to implement them in a flexible and modular manner, and hope that by extending this abstraction level, Qiwi will also be able to deal with compilation strategies, quantum resource estimations, and error correcting codes. The latter are an harder example of meta-algorithms as they involve measurement strategies. To allow quantum algorithms working with them, we will need more than ever, a flexible and modular implementation of them.

At QuantFi, we are faced with client use cases that span the whole financial technology spectrum, from option pricing to machine learning. As QuantFi positions itself as a key provider of quantum software for Finance, we need to be able to quickly deliver prototypes that propose a *quantum solution* to these classical use cases. For instance, when faced with the challenge of harnessing the power of quantum machine learning for credit scoring, one need to use QRAM (a quantum algorithm that loads classical data into a quantum computer), create Gibbs states, and make use of quantum linear algebra. We found that neither QISKIT TERRA nor other quantum computing libraries fulfills our needs since they focus more on low-level structures than the higher-level ones needed for designing quantum algorithms, or that they are restricted to a particular domain.

This is the reason why QuantFi decided to invest into the development of QIWI, a complete, modular, interdisciplinary, state-of-the-art Python library that can be seen as an important drop-in replacement for QISKIT TERRA, with many high-level quantum algorithms ready to be used by engineers and scientists that work in the quantum ecosystem around the world, far beyond Finance. QIWI is based on the low-level components of QISKIT: mainly the quantum circuits machinery and the various backends available. In QIWI, quantum algorithms are expressed as *gates*, making the whole library very expressive and close to how researchers design their own quantum algorithms. QIWI main modules are:

- The Core module contains fundamental gates and algorithms such as quantum boolean oracles, unary and binary encodings (including

QRAM), amplitude amplification and estimation (in which any quantum algorithm can be plugged-in, once represented as a gate itself),

- The `Optimization` module contains not only VQE and QAOA but also *mixed binary optimizers* (MBO), and quantum second-order cone programming (SCOP) for replacing linear and quadratic programming,

- The `LinearAlgebra` module contains the latest HHL algorithm for solving linear systems on a quantum computer, together with hybrid algorithms that are designed for the near term.

Other modules include time evolution (including Hamiltonian simulation), machine learning (OLS, SVM) and various circuit simulators, including a symbolic one for debugging parametrized quantum circuits more easily.

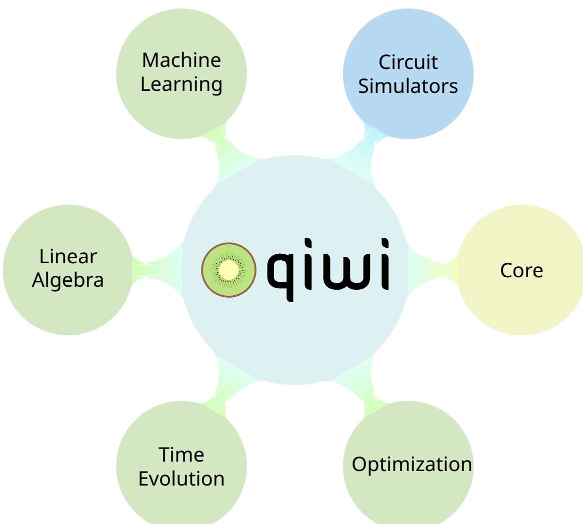

Figure 2: An overview of the different modules composing Qiwi.

Last but not least, QIWI is fully tested and docucmented, making it attractive and user-friendly. In the long-term, QIWI will support its own quantum emulator based on OpenCL, and will be fully hardware agnostic (both classical and quantum). We also seek to include compilation strategies, ancillary qubits processing and gate counting to estimate and optimize the resources required by quantum algorithms. QuantFi plans to release QIWI with an open source license in order to make it available for the whole quantum computing community. Stay tuned for our first release!