QAL Abstract

QAL (Quantum Algorithms Lab) is an innovative visual interactive web app for teaching and researching quantum algorithms and quantum informatics as well as related mathematical fields (such as linear algebra and group theory).

As a teaching tool, QAL targets STEM school students and undergrads of computer science and mathematics as its main users. As a research tool, researchers and graduate students in various scientific fields can strongly benefit from QAL. In addition to quantum computing, the math nicely presented in QAL has applications in many mathematical sciences such as machine learning, computer graphics, cyber-security, and mathematical physics.

QAL in Teaching

An algorithm, also sometimes called a *classical* algorithm, is a sequence of classical steps or operations (a.k.a., moves or actions) — basic operations, like AND, OR, NOT, XOR, or composite ones, like ADD, MULT, and so on.

A *reversible* algorithm is a sequence of (basic or composite) reversible operations — ones like NOT, SWAP, CNOT, CCNOT, CSWAP, and so on.

A *quantum* algorithm is a sequence of (basic or composite) quantum operations — ones like NOT, NEG, CNOT, CCNOT, CSWAP, CNEG, HAD (a.k.a., X, Z, CX, Toff, Fred, CZ, and H, respectively), QFT, and so on.

Now, if a teacher talks to some potential students (e.g., of quantum programming) about complex numbers, then talks to them about matrices and vectors in general then about matrices and vectors of complex numbers in particular, then about the composition or multiplication of sequences of such matrices and vectors ... very likely the students will feel overwhelmed, and may think of mathematical models of quantum algorithms as complex, too abstract, and probably non-intuitive. Most such students will, most likely, feel disconnected from quantum algorithms and quantum informatics.

BUT if the teacher were to present to the students some paintings and drawings that can be "merged," or present them with some pipes or threads that can be connected, or some other everyday objects that students can easily interact with, the students will feel more at home and will think of quantum algorithms as being somewhat similar to concrete objects they are already familiar with, ones that — like pieces of Lego™ — they can interact and "play" with readily.

... and that's precisely what QAL tries to do ... so let's see how QAL does it, shortly after we discuss, in brief, how QAL can also be used in research.

QAL in Research

Quantum Computing (QC) is an important emerging technology that has huge potential, and QAL is a potentially huge and potentially useful software that ambitiously aims to help QC reach its potential.

QC is a vast topic, because it is connected to various branches of science including physics, computer science, informatics, and mathematics.

Like traditional computing, QC has two components: a hardware component and a software component. Unlike regular/traditional/classical computing, QC has not become a reality so far, and has thus not reached its full potential yet, for two main reasons: lack of large-scale quantum computers (a hardware challenge), and lack of many quantum computing applications (a software challenge. QC so far is proven to solve only very few practical computing problems more efficiently than classical computing does, e.g., factoring whole numbers useful in secure communication, searching unstructured data, and simulating certain chemical and physical systems).

As a visual and interactive "virtual lab" the Quantum Algorithms Lab (QAL) can function as a quantum algorithms research assistant (QA RA) that can help in addressing the algorithmic software challenges of QC, by helping students and researchers understand and develop intuitions for the power (and limitations) of quantum algorithms.[1]

Let's see, in an active hands-on tutorial, how QAL can help achieve that lofty goal too. (QAL currently has a rudimentary User Guide, accessible from QAL Help. Posting few QAL Tutorial Videos online is intended in the near future.)

QAL Overview

QAL helps its user explore quantum algorithms and related linear algebra concepts by presenting a number of standard and novel diagrams or *visuals* that represent a quantum algorithm (as a sequence of matrices). These visuals constitute the main QAL Presentations section of QAL. (See the 'QAL Lite Screenshots' file.)

The QAL user can directly interact with these visuals, and can use them (or use the QAL Dashboard, in the foldable section of QAL) to modify the steps of their quantum algorithm and *immediately* (in near real-time) see the effect of the changes they make on the overall operation of the algorithm.

QAL as such helps significantly improve understanding linear algebra and group theory, and understanding the subject matters (such as quantum algorithms) that are modeled by the vectors and matrices that QAL visually presents.[2]

[1] Among the subareas of quantum algorithms research that QAL has an inherent affinity for is analyzing the eigenvectors and eigenvalues — known as *spectrum* — of (efficient) quantum operations — using tools such as ones used in Spectral Graph Theory (SGT) — as well as visually and interactively studying the recursive and self-similar structure of these operations.

[2] Two simple linear algebra/quantum algorithms observations that QAL allowed making and confirming are that: (1) following a NOT operation with two applications of a QFT

QAL Code Components

As a software program, QAL's source code is composed of two parts that have distinct responsibilities in the app and that are maintained largely separately. The main QAL user interface, including most of the QAL Dashboard, and few (currently 3) basic interactive visuals constitute **QAL Lite**, the first part of QAL. A number (currently 5) of new advanced interactive visuals, together with few Dashboard dials and controls, constitute the second part of QAL, collectively named the Quantum Algorithms/Linear Algebra Novel Interactive Visuals (**QALA NIVs**).

Both QAL Lite and QA LA NIVs have current basic implementations, and both can be significantly developed further (see below). QAL Lite is a copyrighted open-source software that is available for use for free online. Since the QAL team is seeking to patent QA LA NIVs at the USPTO (United States Patent and Trademark Office) as novel software technologies, more details on QA LA NIVs are currently available only after signing an NDA (Non-Disclosure Agreement).

The relation between QAL, QAL Lite, the basic interactive visuals (in QAL Lite), and QA LA NIVs (patent pending) can probably be best explained via an analogy—an analogy that likens interactive visuals to engines/motors. In such an analogy, QAL Lite is like a basic vehicle (e.g., a car, train, or an airplane) with a basic engine (corresponding to the basic interactive visuals). Without any interactive visuals, QAL and QAL Lite are merely like the vehicle's frame & body that, without "engines" (i.e., interactive visuals), cannot move. Adding QA LA NIVs to QAL Lite (to form QAL—i.e., QAL = QAL Lite + QA LA NIVs) is like replacing the basic engine (of QAL Lite) with—or adding to it—a more powerful engine (i.e., QA LA NIVs), which extends the power and reach of the vehicle (corresponding to 'the pedagogic and research value of QAL in exposing the structure of quantum operations/complex matrices') by 10-fold or more (!). In such an analogy, QAL Lite is like a basic car (with a basic engine), while QAL is like a sports car (with a much more advanced and powerful engine).

QAL Lite Presentations

The three basic interactive visuals currently available in QAL Lite present a quantum algorithm as: (1) a sequence of heatmaps and a result/overall heatmap, (2) a sequence of complex matrices, in textual form, and a result/overall matrix, and (3) a sequence of circuit blocks, i.e., quantum steps in familiar quantum circuit diagrams. A visualization of quantum states (i.e., state vectors) before and after each step of the algorithm can be displayed or turned off. (Screenshots of

operation is equivalent to the 'add 1 with wraparound' — a.k.a., rotate clockwise — operation, and (2) following two applications of QFT with a NOT is equivalent to the 'subtract 1 with wraparound' — a.k.a., rotate counterclockwise — operation. Expressed algebraically, QAL allowed easily confirming that

$$QFT \circ QFT \circ X = ADD1 (= RotCW) \text{ and } X \circ QFT \circ QFT = SUB1 (= RotCCW).$$

(See the 'QAL Lite Screenshots' file for visual confirmation of these and other similar facts.)

QAL Lite can be found in the accompanying 'QAL Lite Screenshots' document available online.)

Similar Software

QAL strongly adopts a visual interactive approach to mathematics. Like-minded software includes popular apps such as Desmos, which uses interactive and animated visuals and a foldable dashboard to teach basic high school math (www.desmos.com). A much earlier software (for desktop PCs) that adopted a similar approach was Geometer. GeoGebra is a more recent software similar to Geometer that's available for all major platforms.

Many desktop and smartphone apps have been developed that attempt to present quantum programming and the construction of quantum algorithms as a game. These apps include Meqanic (for iOS), QLogic (for Android), and IBM's Hello Quantum.

Also, many web apps exist that attempt to present quantum theory and quantum software visually (and interactively). (More on these web apps, and a comparison with QAL, are planned for inclusion in future versions of this document.)

Other software bearing some similarities to QAL includes mobile, desktop, and web apps and games built using R, Julia, Mathematica, SageMath, Matlab, Python and other similar platforms and frameworks for developing visual interactive mathematical software (such apps have gained popularity in various subfields of computer science, e.g., Data Science, Artificial Intelligence/AI, and High-Performance Computing/HPC).

For more information on QAL, kindly reach out to `QAL@rice.edu`, `moez.qal@srtacity.sci.eg`, or `quantum.computing@rice.edu`.

QAL Lite is freely available online, together with basic documentation, sample screenshots, and few basic tutorial videos (to be available soon), at `https://q-info.github.io/QAL-Lite/` and at `http://eng.staff.alexu.edu.eg/staff/moez/QAL/Lite/` (sometimes down/inactive).

Joining The QAL Team

QAL is currently in its early development phases (*i.e.*, an alpha software). The latest version of QAL is QAL 0.9.

QAL Lite can be developed further in multiple ways, including either by:

1. testing, speeding up, and improving the current codebase of QAL,
2. adding ZX diagrams (zxcalculus.org), as a basic interactive visual, to the set of QAL Lite presentations of quantum algorithms,
3. adding more features to QAL's user interface (e.g., activating and expanding the QAL Game Mode), or

4. porting QAL to other platforms (e.g., smartphones and other handheld devices). (As a portable web app, QAL runs consistently across all computing devices having modern web browsers — including, e.g., running on the most recent versions of Firefox and Chrome on Android smartphones. Developing QAL as a native application of other platforms is nevertheless desirable for performance and familiarity-of-UI reasons.)

Also, QA LA NIVs currently include five fully implemented novel interactive visuals. Few more novel interactive visuals are currently under active development for inclusion in future versions of QAL. Several novel advanced interactive visuals can be further added to QA LA NIVs.

To support the future development of QAL — i.e., of QAL Lite and QA LA NIVs — the QAL team is currently seeking new team members. Highly-qualified and highly-motivated software developers, especially ones with expertise in game design, in 3D computer graphics, or in VR and AR apps (WebXR) development, are particularly encouraged to join the QAL team.