A Quantum Computer Simulator
Version 0.9beta
— Documentation —

Andreas de Vries
FH Südwestfalen University of Applied Sciences, Haldener Straße 182, D-58095 Hagen, Germany
e-mail: de-vries@fh-swf.de

Contents

1 About this Program

The program `jQuantum` is a quantum computer simulator. It simulates the implementation of quantum circuits on a small quantum register up to about 15 qubits. Its main intention is to *create images* — images which may help to learn and understand quantum circuits, and which perhaps will serve as building blocks for inventing new quantum algorithms.

2 Short manual

2.1 Installation

You need the Java Virtual Machine JRE 1.4.1 or higher to run `jQuantum`. If you do not have it, you can simply download it from http://java.sun.com. To start jQuantum, under most platforms you simply click

on the jar-file. If this does not work you can start the shell, change into the directory of `jQuantum.jar` and and type in: `java -jar jQuantum.jar`

2.2 The structure of the work space window

After having started the program, you see the work space window (Fig. 1). The window is structured as

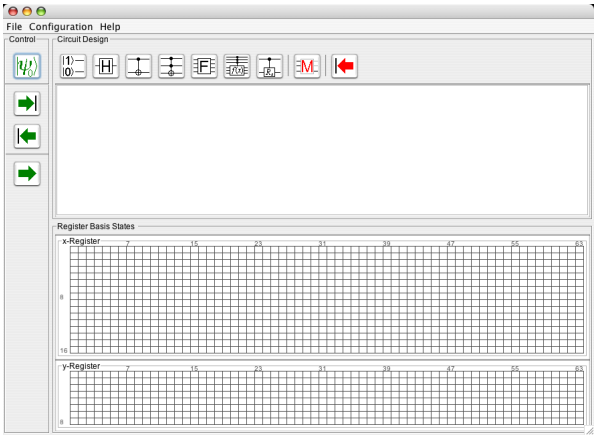

Figure 1: The initial work space window.

follows. On the left part you find the control panel, in which there are buttons to initialize the design and to control the run of the quantum program. The upper horizontal panel consists of buttons with which you can add quantum gates to the circuit design canvas directly below the panel. At the bottom you find a representation of the quantum register. It is divided in a part which receives the input and yields the output, called the *x*-register, and into a part called *y*-register, which serves mainly as a temporary qubit storage, as, e.g., for Shor's function evaluation principle.

2.3 How to initialize a quantum register

The first thing you have to do is to click on the upper left button in the control panel, marked by $|\psi_0\rangle$. With it, the number of qubits for each register (*x* or *y*) is determined; each qubit is input into the circuit as a wire. You should not choose more than about 15 qubits in total.

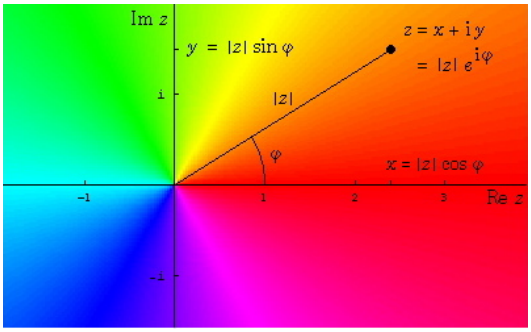

Figure 2: The color map of the complex plane. The hue depends uniquely on the phase φ. (The hue of the origin $z = 0$ remains undetermined, we define it here as black). For instance, a positive real number is red, a negative one is light blue [7, 1]

A qubit register of size n is determined by 2^n basis states. In the register panel of the work place window, each basis state is represented as a single square whose (complex) amplitude z is given by its color according to the color map (Fig. 2). If a qubit $|j\rangle$ has the real amplitude $a \in (0, 1]$, then the j-th square in the register panel is red. This is the usual (classical) initial value of a qubit, if it is set. All qubits which have a vanishing amplitude $z = 0$ are colored black.

By default, the initial qubit state is $|0 \ldots 0\rangle$. To change the initial state, and also to reinitialize the quantum circuit during or after a run, you can press the button $|1\rangle$ $|0\rangle$.

2.4 Design a circuit

After having initialized the quantum register, you can build in several quantum gates by simply clicking on the buttons in the circuit design panel. Usually you are prompted to determine the qubits on which the gate shall act upon (Fig. 3).

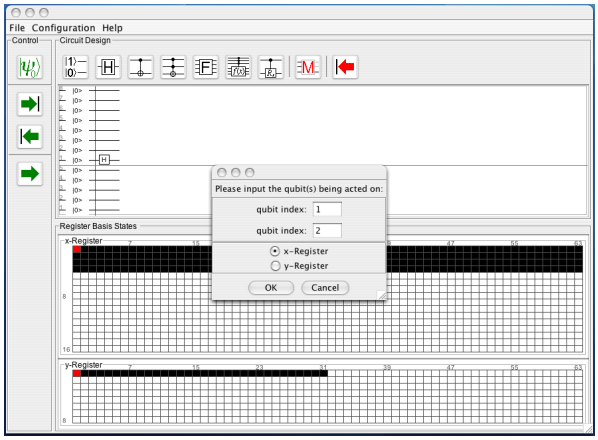

Figure 3: To build in a quantum gate, click on a button in the circuit design panel. You are then prompted to add further informations.

The left button is to initialize the input qubits, the gate **M** denotes a measurement, and with the red arrow button ← on the right you can delete the currently last quantum gate built in the register.

2.5 Run the algorithm

There are two modes in which you can run the algorithm implemented by the quantum circuit. You can either go stepwise by clicking the two green "blocked" arrows →| and |← on the left control panel; or you can perform the whole algorithm by clicking on the green arrow → lower-most in the control panel (Fig. 4). The current position in the circuit is indicated by a vertical green line int the circuit canvas. The current quantum register state is displayed synchronously in the regiter panel at the bottom of the window.

If you want to restart the algorithm, you can press the button $|1\rangle$ $|0\rangle$. With it, you can also change the initial qubits.

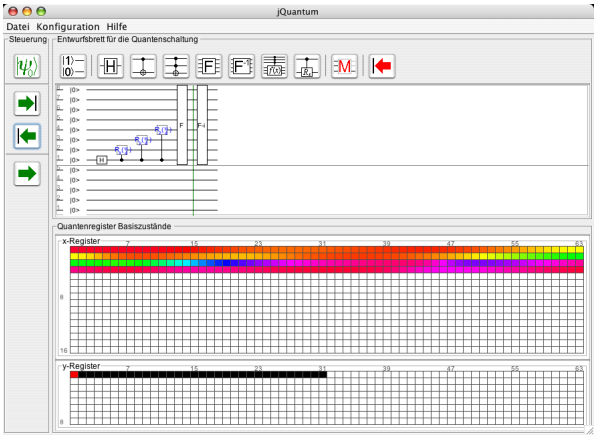

Figure 4: Running a quantum program.

3 Theoretical Background

3.1 Why quantum computers?

There are several reasons why quantum computers should be studied. First, quantum computation soon will become a revolutionary *new technology*. Quantum computers will be *extremely* faster than the classical computers of our time. New technologies always offer new perspectives, not only economically. They bind the best intellectual resources of a society and are the key for sustained growth of wealth and science.

One of the most important consequences that runnable quantum computers with a register of about 1 kq (1000 qubit) is that all current public key systems will be cracked![1] OK, this may last one or more decades from now, but quantum computation is a problem (as well as a solution, by the way, since it offers new cryptography!) coming with certainty to e-commerce firms and security services. It is an economical and a political factor of high relevance.

Another reason concerns scientific and theoretical aspects. Quantum registers are among the simplest many-particle quantum systems, being describable as "spinors". So, quantum computers could serve as laboratories to investigate quantum effects. Especially one feature can be studied extensively, because quantum computation is based mainly upon it: "quantum entanglement". Sloppily said, two distinguished particles can be brought into a common superposition state called entanglement which has the strange property that if a property of one of the particle is measured (e.g., its spin), then the other has the opposite value, no matter how far both are apart; but until the measurement, the value of the property might be completely undetermined, but then they are "anti-" synchronized. This phenomenon had been rather controversial since it discovery by Einstein and his coworkers Podolsky and Rosen [3], although they thought to have found an impossible consequence of quantum mechanics and thus a counterexample to falsify this theory. However, complex as Nature is, the phenomenon could be demonstrated experimentally in the 1980's. Nowadays, nearly no one doubts the reality of entanglement, but the philosophical implications are still not clear.

[1] The reason is that since 1994, when Peter Shor published it, a quantum algorithm is known which can factorize big numbers and solve the "discrete logarithm" in an *efficient* way (i.e., with polynomial complexity). Although a quantum algorithm which cracks a symmetric key efficiently has not been invented to date (at least to my knowledge), I suppose that it will be a question of a few years until one appears. (Perhaps one of the cornerstones of such an algorithm is already found with the "hidden subgroup".)

3.2 What is quantum computation?

Whereas classical information is based on *bits* taking only two possible values, quantum information relies on *qubits*: a qubit generalizes the concept of a bit since it may not only attain two values $|0\rangle$ and $|1\rangle$, say, but also "superpositions" of them, i.e., vector sums like $|0\rangle + |1\rangle$ or $|0\rangle - |1\rangle$. This is a radically new logical property, totally unknown in classical information processing. It enables new kinds of logical gates and algorithms, working on *quantum registers*, ordered collections of several qubits.

Although the computation of a whole quantum circuit or algorithm can be highly nonclassical, both input and output must be classical information, encoded by bits. This is obvious because we want algorithms to solve questions that we have, and we have them "classically", and we need information in a classical form since this only answers our question.[2] To receive the output, the quantum register, or parts of it, has to be *measured*. Measurement is the only way to receive classical information from quantum systems.

To summarize, a quantum computer has bits as input, it works with qubits, and gives bits as output.

3.3 Quantum logic

The basis for quantum computation is not Boolean logic, but *quantum logic*. To date, there is still no appropriate quantum-logical calculus comparable to the classical calculus based on Boolean algebra. There are at least two essential differences between quantum and Boolean logic. One is that any quantum gate has to be *reversible*, i.e., input and output must always correspond uniquely to one another. In particular, the number of input and output qubits have to be equal. This is different than in the Boolean case, where most gates have two input bits and only one output bit. In fact, all basic binary operations of Boolean algebra ($\wedge$, $\vee$, XOR, NAND, NOR, ...) are 2-1 valued, which implies that they are not reversible: in fact, since $1 \wedge 0 = 0 \wedge 1 = 0 \wedge 0 = 0$, you cannot deduce from the result "0" which values the input bits have had.[3]

Another difference between quantum and Boolean logic is that quantum gates can transform a qubit basis, say $\{|0\rangle, |1\rangle\}$, to another, for instance $\{|0\rangle + |1\rangle, |0\rangle - |1\rangle\}$, just as a vector basis can be changed by reflections or rotations. This property is impossible in Boolean logic, where any operation transforms to one of the two values 0 or 1. In other words, the basis is never changed in Boolean logic.

3.4 Where can I get more information?

One of the best references concerning quantum computation to date is certainly the textbook of Nielsen and Chuang [6]. It covers nearly all branches of the wide subject and gives good introductions to each of them; often, the style is visionary and inspiring, although the contents is not trivial. Another very readable reference is the AMS proceeding edited by Lomonaco [5]. It contains brief introductions and survey to different aspects of quantum computation, often supplemented by illustrative examples.

A website where you can get actual informations is the homepage of the *Centre for Quantum Computation*,

www.qubit.org

[2] Perhaps in future times we will have inherently quantum-mechanical questions or only need quantum information as answers. But this results in a whole new class of *problems* and goes beyond this project.

[3] However, already in 1961 Rolf Landauer [4] from IBM, and later Bennett [2], have shown that any classical computation can equivalently be implemented in a reversible manner. Therefore, quantum logic in fact is a proper generalization of Boolean logic.

Here you also find online tutorials on various subjects, as well as links to universities and institutes working and researching in quantum computation.

A reference concerning the simulation of a quantum computer is the web address

www.enyo.de/libquantum/

It contains the open source C library `libquantum`.

References

[1] Color traingle CIE 1931, http://www.efg2.com/Lab/, Gernot Hoffmann: *CIE Color Space*, www.fho-emden.de/ hoffmann/colcie290800.pdf, Fred Bunting: *The ColorShop Color Primer*, www.xrite.com/documents/mktg/ColorPrimer.pdf

[2] Charles H. Bennett. 'Logical reversibility of computation'. *IBM J. Research and Development*, **17**, 525–532 (1973).

[3] Albert Einstein, B. Podolsky, and Nathan Rosen. 'Can quantum mechanical description of physics reality be considered complete?'. *Phys. Rev.*, **47**, 777–780 (1935)

[4] Rolf Landauer. 'Irreversibility and heat generation in the computing process'. *IBM J. Research and Development*, **3**, 183–191 (1961)

[5] Samuel Lomonaco Jr. *Quantum Computation. A Grand Mathematical Challenge for the 21st Century and the Millenium*. Proceedings of the Symposia of Applied Mathematics, vol. 58, American Mathematical Society, Providence, Rhode Island, 2002.

[6] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2000.

[7] Bernd Thaller. *Visual Quantum Mechanics*. Springer-Verlag, New York Heidelberg, 2000. http://www.kfunigraz.ac.at/imawww/vqm/.