Projects | Quantum Information Theory

The Delft Experiment

In 1964, John Bell proved that theories compatible with naïve ideas of locality and determinism can only produce correlations weaker than an upper bound – now known as Bell's inequality – and that quantum mechanics predicts correlations stronger than that [1]. Actually producing these strong correlations in a laboratory was, for decades, an insurmountable problem: the best attempts needed an additional assumption – the *fair-sampling* assumption – that, although physically reasonable, made the experimental setups useless for the challenging task of quantum cryptography based on these correlations [2]. The race came to an end in 2015, when three experimental groups reported success [3–5], the first being our neighbours in Delft.

In this project you must be able to prove Bell's theorem, explain what the fair-sampling assumption is, and why it poses a problem for quantum cryptography. Additionally, you need to analyse the experimental data from the Delft experiment, which is available here [6]. You should check it for violation of the Bell inequality, space-like separation, and faster-than-light communication.

As an informal introduction to Bell's theorem, I suggest my own blog [7]. For the technical part, I recommend the Dutch paper [3] and its Supplementary Information [8].

[1] J. S. Bell, "On the Einstein-Poldolsky-Rosen paradox", Physics **1**, 195–200 (1964).

[2] Z. Merali, *Quantum Mechanics Braces for the Ultimate Test*, (2011) http://science.sciencemag.org/content/331/6023/1380.full.

[3] B. Hensen et al., "Experimental loophole-free violation of a Bell inequality using entangled electron spins separated by 1.3 km", Nature **526**, 682–686 (2015), arXiv:1508.05949 [quant-ph].

[4] M. Giustina et al., "Significant-Loophole-Free Test of Bell's Theorem with Entangled Photons", Phys. Rev. Lett. **115**, 250401 (2015), arXiv:1511.03190 [quant-ph].

[5] L. K. Shalm et al., "Strong Loophole-Free Test of Local Realism*", Phys. Rev. Lett. **115**, 250402 (2015), arXiv:1511.03189 [quant-ph].

[6] B. Hensen et al., https://data.4tu.nl/repository/uuid:6e19e9b2-4a2d-40b5-8dd3-a660bf3c0a31.

[7] M. Araújo, *Understanding Bell's theorem*, (2016) http://mateusaraujo.info/2016/07/15/understanding-bells-theorem-part-1-the-simple-version/.

[8] B. Hensen et al., *Experimental loophole-free violation of a Bell inequality using entangled electron spins separated by 1.3 km – Supplementary Information*, http://www.nature.com/nature/journal/v526/n7575/extref/nature15759-s1.pdf.

Shor's Algorithm

Shor's algorithm, introduced in 1994 by Peter Shor [1], is the most dramatic demonstration of the power of a quantum computer. It can factor b-bit integers in time $O(b^3)$, whereas the most powerful classical algorithm available, the general number field sieve [2], needs time that grows exponentially with b. This exponential speed-up is enough to break all public-key cryptosystems in use nowadays – RSA, Diffie–Hellman, and Elliptic-curve – and thus has wide-ranging security implications.

In this project, you should implement Shor's algorithm in a classical computer – as currently existing quantum computers can only run Shor's algorithm for trivial inputs. You should both implement the whole algorithm classically, and do a hybrid implementation where the crucial order-finding subroutine is done by directly simulating a quantum circuit. One should compare these implementations, and check what is the number of bits of the largest integers one can factor in both cases.

Reference [1] is only of historical interest; one should first read an informal introduction to Shor's algorithm by Scott Aaronson [3], and then use Nielsen & Chuang's book as a guide for the actual implementation [4].

[1] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer", SIAM J.Sci.Statist.Comput. **26**, 1484 (1997), arXiv:quant-ph/9508027.

[2] Wikipedia, *General number field sieve*, https://en.wikipedia.org/wiki/General_number_field_sieve.

[3] S. Aaronson, *Shor, I'll do it*, (2007) https://www.scottaaronson.com/blog/?p=208.

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

IBM's Quantum Computer

IBM has two prototype quantum computers, one with 4 and another with 5 qubits, available for experimentation through the internet [1]. In this project one should use them to run simple quantum circuits that we have learned throughout the course; the precise choice of the circuits is free, but must include at a minimum Grover's algorithm for an 8-item database and quantum state tomography for a single qubit.

You should use Nielsen & Chuang's book as a general reference [2], and IBM's user guide for how to implement circuits in their computer [3].

[1] IBM, *IBM Quantum Experience*, https://quantumexperience.ng.bluemix.net/qx/community.

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

[3] IBM, *Full User Guide*, https://quantumexperience.ng.bluemix.net/qx/tutorial?sectionId=full-user-guide&page=introduction.