Appendix A
QY

In the course of the experimental work described in this thesis we have developed a broad general-purpose base of computer code (`qy`), which is maintained and documented as a library to encourage re-use. The majority of this code is written in the `Python` programming language,[1] with some compiled extensions written in `C` or `Cython` for speed. Both of these languages are free and open source.

`qy` includes modules for data acquisition (DAQ) and hardware control, data logging and analysis, and numerical simulation, with a specific emphasis on tasks which often occur in experimental quantum photonics. The code is currently open source, and can be obtained via `git`:

`https://github.com/peteshadbolt/qy`

The top-level structure of the library is as follows:

- `qy.analysis`: Various standard metrics and tools for data analysis.
- `qy.formats`: File formats, in particular an efficient format to represent multi-photon coincidence-counting data.
- `qy.graphics`: Utility functions for graphics and plotting.
- `qy.hardware`: Interfaces to various pieces of standard laboratory apparatus, including FPGA counting systems, the DPC-230 described in Sect. 6.2, Toptica diode lasers, Coherent Ti:Saph lasers, custom powermeters, Thor labs SMC100 power meters, silica-on-silicon thermal phase shifters, etc.
- `qy.settings`: Utility functions to read, write and persist global settings.
- `qy.simulation`: Provides general quantum information primitives, including single-qubit states and operators, frequently used two-qubit states and gate operations, measures such as quantum state fidelity and concurrence, a circuit-model simulator, and an optimized linear-optics simulator, capable of calculating multi-photon states and statistics in arbitrary linear optical networks.

[1] `http://www.python.org/`.

P. Shadbolt, *Complexity and Control in Quantum Photonics*,
Springer Theses, DOI 10.1007/978-3-319-21518-1

- qy.util: Utility functions.
- qy.wx: Extends the functionality of the wx GUI library.

Here we will discuss two components in particular: the linear_optics simulation package and the .counted file format.

A.1 Universal Linear Optics Simulator

The module qy.simulation.linear_optics provides a simple means to simulate multiphoton states and statistics in arbitrary linear optical circuits. This work draws upon ideas and code kindly provided by Jasmin Meinecke, Nick Russell, Jacques Carolan. The numerical method is exactly that described in Sect. 1.5.3, and as such depends almost entirely on the calculation of permanents. We have developed optimized code to compute the permanent of complex matrices, using a number of different algorithms and implementations. We implemented the core algorithm using Cython, a compiled language which can typically achieve much better performance than standard Python, which is interpreted rather than compiled. Typical real-world performance of these methods is summarized in Fig. A.1. The library is very easy to use:

```
import numpy as np
from qy.simulation import linear_optics as lo

# Load up a device from a JSON definition file:
device=lo.beamsplitter_network(json='devices/cnot_mz.json')
print device
print device.get_unitary().round(2)
print device.nmodes

# Draw the waveguide structure as a PDF file
device.draw('devices/cnot_mz.pdf')

# Make a simulator, and link it to the device
simulator=lo.simulator(device, nphotons=2)

# Print out the basis
print simulator.basis

# Set the input state to two photons in the top mode, and look at
# the output probabilities and output state
simulator.set_input_state([0, 0])
print simulator.input_state
print simulator.get_probabilities().round(2)
print simulator.get_output_state()

# Superposition input states, and classical statistics
state=simulator.basis.get_state()
state[0,1]=1/np.sqrt(2)
state[3,4]=1/np.sqrt(2)
print state
simulator.set_input_state(state)
simulator.set_visibility(0.5)
print simulator.get_probabilities()
```

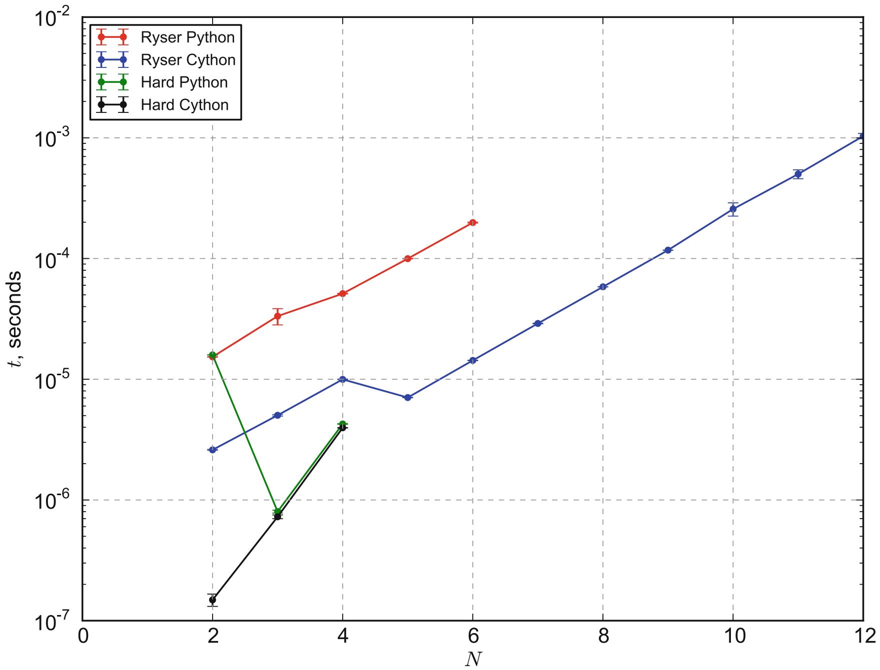

Fig. A.1 Estimated performance of various implementations of algorithms to compute the permanent, tested against 1000 Haar-random $N \times N$ matrices. *Red* and *blue lines* show average execution times for Ryser's algorithm, implemented in Python and Cython respectively, as a function of N. *Green* and *black lines* correspond to execution times for hard-coded implementations *up* to $N = 4$, again in Python and Cython respectively (color online)

```
# Performance test: 4 photons in 16 modes of a Haar-random U
# Hilbert space dimension is now 3876
device=lo.random_unitary(16)
simulator=lo.simulator(device, nphotons=4)
simulator.set_input_state(range(4))  # Photons go in the top 4 modes
probs=simulator.get_probabilities(label=True)
```

When computing the permanent, it was noticed (by Nick Russell) that hard-coded routines can give a significant advantage in speed for small matrices, as the overhead associated with loops and conditional statements can be completely avoided. For completeness we include code up to $N = 4$, beyond which the advantage with respect to Ryser's algorithm is negligible.

```
def perm_2x2(a):
    """ An explicit 2x2 permanent """
    return a[0,0]*a[1,1]
        + a[1,0]*a[0,1]

def perm_3x3(a):
    """ An explicit 3x3 permanent """
    return a[0,0]*a[1,1]*a[2,2]
        + a[0,0]*a[2,1]*a[1,2]
        + a[1,0]*a[0,1]*a[2,2]
        + a[1,0]*a[2,1]*a[0,2]
        + a[2,0]*a[0,1]*a[1,2]
        + a[2,0]*a[1,1]*a[0,2]
```

```python
def perm_4x4(a):
    """ An explicit 4x4 permanent """
    return a[0,0]*a[1,1]*a[2,2]*a[3,3]  + a[0,0]*a[1,1]*a[3,2]*a[2,3]
        + a[0,0]*a[2,1]*a[1,2]*a[3,3]  + a[0,0]*a[2,1]*a[3,2]*a[1,3]
        + a[0,0]*a[3,1]*a[1,2]*a[2,3]  + a[0,0]*a[3,1]*a[2,2]*a[1,3]
        + a[1,0]*a[0,1]*a[2,2]*a[3,3]  + a[1,0]*a[0,1]*a[3,2]*a[2,3]
        + a[1,0]*a[2,1]*a[0,2]*a[3,3]  + a[1,0]*a[2,1]*a[3,2]*a[0,3]
        + a[1,0]*a[3,1]*a[0,2]*a[2,3]  + a[1,0]*a[3,1]*a[2,2]*a[0,3]
        + a[2,0]*a[0,1]*a[1,2]*a[3,3]  + a[2,0]*a[0,1]*a[3,2]*a[1,3]
        + a[2,0]*a[1,1]*a[0,2]*a[3,3]  + a[2,0]*a[1,1]*a[3,2]*a[0,3]
        + a[2,0]*a[3,1]*a[0,2]*a[1,3]  + a[2,0]*a[3,1]*a[1,2]*a[0,3]
        + a[3,0]*a[0,1]*a[1,2]*a[2,3]  + a[3,0]*a[0,1]*a[2,2]*a[1,3]
        + a[3,0]*a[1,1]*a[0,2]*a[2,3]  + a[3,0]*a[1,1]*a[2,2]*a[0,3]
        + a[3,0]*a[2,1]*a[0,2]*a[1,3]  + a[3,0]*a[2,1]*a[1,2]*a[0,3]

def perm_5x5(a):
    """ An explicit 5x5 permanent """
    return a[0,0]*a[1,1]*a[2,2]*a[3,3]*a[4,4]  + a[0,0]*a[1,1]*a[2,2]*a[4,3]*a[3,4]
        + a[0,0]*a[1,1]*a[3,2]*a[2,3]*a[4,4]  + a[0,0]*a[1,1]*a[3,2]*a[4,3]*a[2,4]
        + a[0,0]*a[1,1]*a[4,2]*a[2,3]*a[3,4]  + a[0,0]*a[1,1]*a[4,2]*a[3,3]*a[2,4]
        + a[0,0]*a[2,1]*a[1,2]*a[3,3]*a[4,4]  + a[0,0]*a[2,1]*a[1,2]*a[4,3]*a[3,4]
        + a[0,0]*a[2,1]*a[3,2]*a[1,3]*a[4,4]  + a[0,0]*a[2,1]*a[3,2]*a[4,3]*a[1,4]
        + a[0,0]*a[2,1]*a[4,2]*a[1,3]*a[3,4]  + a[0,0]*a[2,1]*a[4,2]*a[3,3]*a[1,4]
        + a[0,0]*a[3,1]*a[1,2]*a[2,3]*a[4,4]  + a[0,0]*a[3,1]*a[1,2]*a[4,3]*a[2,4]
        + a[0,0]*a[3,1]*a[2,2]*a[1,3]*a[4,4]  + a[0,0]*a[3,1]*a[2,2]*a[4,3]*a[1,4]
        + a[0,0]*a[3,1]*a[4,2]*a[1,3]*a[2,4]  + a[0,0]*a[3,1]*a[4,2]*a[2,3]*a[1,4]
        + a[0,0]*a[4,1]*a[1,2]*a[2,3]*a[3,4]  + a[0,0]*a[4,1]*a[1,2]*a[3,3]*a[2,4]
        + a[0,0]*a[4,1]*a[2,2]*a[1,3]*a[3,4]  + a[0,0]*a[4,1]*a[2,2]*a[3,3]*a[1,4]
        + a[0,0]*a[4,1]*a[3,2]*a[1,3]*a[2,4]  + a[0,0]*a[4,1]*a[3,2]*a[2,3]*a[1,4]
        + a[1,0]*a[0,1]*a[2,2]*a[3,3]*a[4,4]  + a[1,0]*a[0,1]*a[2,2]*a[4,3]*a[3,4]
        + a[1,0]*a[0,1]*a[3,2]*a[2,3]*a[4,4]  + a[1,0]*a[0,1]*a[3,2]*a[4,3]*a[2,4]
        + a[1,0]*a[0,1]*a[4,2]*a[2,3]*a[3,4]  + a[1,0]*a[0,1]*a[4,2]*a[3,3]*a[2,4]
        + a[1,0]*a[2,1]*a[0,2]*a[3,3]*a[4,4]  + a[1,0]*a[2,1]*a[0,2]*a[4,3]*a[3,4]
        + a[1,0]*a[2,1]*a[3,2]*a[0,3]*a[4,4]  + a[1,0]*a[2,1]*a[3,2]*a[4,3]*a[0,4]
        + a[1,0]*a[2,1]*a[4,2]*a[0,3]*a[3,4]  + a[1,0]*a[2,1]*a[4,2]*a[3,3]*a[0,4]
        + a[1,0]*a[3,1]*a[0,2]*a[2,3]*a[4,4]  + a[1,0]*a[3,1]*a[0,2]*a[4,3]*a[2,4]
        + a[1,0]*a[3,1]*a[2,2]*a[0,3]*a[4,4]  + a[1,0]*a[3,1]*a[2,2]*a[4,3]*a[0,4]
        + a[1,0]*a[3,1]*a[4,2]*a[0,3]*a[2,4]  + a[1,0]*a[3,1]*a[4,2]*a[2,3]*a[0,4]
        + a[1,0]*a[4,1]*a[0,2]*a[2,3]*a[3,4]  + a[1,0]*a[4,1]*a[0,2]*a[3,3]*a[2,4]
        + a[1,0]*a[4,1]*a[2,2]*a[0,3]*a[3,4]  + a[1,0]*a[4,1]*a[2,2]*a[3,3]*a[0,4]
        + a[1,0]*a[4,1]*a[3,2]*a[0,3]*a[2,4]  + a[1,0]*a[4,1]*a[3,2]*a[2,3]*a[0,4]
        + a[2,0]*a[0,1]*a[1,2]*a[3,3]*a[4,4]  + a[2,0]*a[0,1]*a[1,2]*a[4,3]*a[3,4]
        + a[2,0]*a[0,1]*a[3,2]*a[1,3]*a[4,4]  + a[2,0]*a[0,1]*a[3,2]*a[4,3]*a[1,4]
        + a[2,0]*a[0,1]*a[4,2]*a[1,3]*a[3,4]  + a[2,0]*a[0,1]*a[4,2]*a[3,3]*a[1,4]
        + a[2,0]*a[1,1]*a[0,2]*a[3,3]*a[4,4]  + a[2,0]*a[1,1]*a[0,2]*a[4,3]*a[3,4]
        + a[2,0]*a[1,1]*a[3,2]*a[0,3]*a[4,4]  + a[2,0]*a[1,1]*a[3,2]*a[4,3]*a[0,4]
        + a[2,0]*a[1,1]*a[4,2]*a[0,3]*a[3,4]  + a[2,0]*a[1,1]*a[4,2]*a[3,3]*a[0,4]
        + a[2,0]*a[3,1]*a[0,2]*a[1,3]*a[4,4]  + a[2,0]*a[3,1]*a[0,2]*a[4,3]*a[1,4]
        + a[2,0]*a[3,1]*a[1,2]*a[0,3]*a[4,4]  + a[2,0]*a[3,1]*a[1,2]*a[4,3]*a[0,4]
        + a[2,0]*a[3,1]*a[4,2]*a[0,3]*a[1,4]  + a[2,0]*a[3,1]*a[4,2]*a[1,3]*a[0,4]
        + a[2,0]*a[4,1]*a[0,2]*a[1,3]*a[3,4]  + a[2,0]*a[4,1]*a[0,2]*a[3,3]*a[1,4]
        + a[2,0]*a[4,1]*a[1,2]*a[0,3]*a[3,4]  + a[2,0]*a[4,1]*a[1,2]*a[3,3]*a[0,4]
        + a[2,0]*a[4,1]*a[3,2]*a[0,3]*a[1,4]  + a[2,0]*a[4,1]*a[3,2]*a[1,3]*a[0,4]
        + a[3,0]*a[0,1]*a[1,2]*a[2,3]*a[4,4]  + a[3,0]*a[0,1]*a[1,2]*a[4,3]*a[2,4]
        + a[3,0]*a[0,1]*a[2,2]*a[1,3]*a[4,4]  + a[3,0]*a[0,1]*a[2,2]*a[4,3]*a[1,4]
        + a[3,0]*a[0,1]*a[4,2]*a[1,3]*a[2,4]  + a[3,0]*a[0,1]*a[4,2]*a[2,3]*a[1,4]
        + a[3,0]*a[1,1]*a[0,2]*a[2,3]*a[4,4]  + a[3,0]*a[1,1]*a[0,2]*a[4,3]*a[2,4]
        + a[3,0]*a[1,1]*a[2,2]*a[0,3]*a[4,4]  + a[3,0]*a[1,1]*a[2,2]*a[4,3]*a[0,4]
        + a[3,0]*a[1,1]*a[4,2]*a[0,3]*a[2,4]  + a[3,0]*a[1,1]*a[4,2]*a[2,3]*a[0,4]
        + a[3,0]*a[2,1]*a[0,2]*a[1,3]*a[4,4]  + a[3,0]*a[2,1]*a[0,2]*a[4,3]*a[1,4]
        + a[3,0]*a[2,1]*a[1,2]*a[0,3]*a[4,4]  + a[3,0]*a[2,1]*a[1,2]*a[4,3]*a[0,4]
        + a[3,0]*a[2,1]*a[4,2]*a[0,3]*a[1,4]  + a[3,0]*a[2,1]*a[4,2]*a[1,3]*a[0,4]
        + a[3,0]*a[4,1]*a[0,2]*a[1,3]*a[2,4]  + a[3,0]*a[4,1]*a[0,2]*a[2,3]*a[1,4]
        + a[3,0]*a[4,1]*a[1,2]*a[0,3]*a[2,4]  + a[3,0]*a[4,1]*a[1,2]*a[2,3]*a[0,4]
        + a[3,0]*a[4,1]*a[2,2]*a[0,3]*a[1,4]  + a[3,0]*a[4,1]*a[2,2]*a[1,3]*a[0,4]
        + a[4,0]*a[0,1]*a[1,2]*a[2,3]*a[3,4]  + a[4,0]*a[0,1]*a[1,2]*a[3,3]*a[2,4]
        + a[4,0]*a[0,1]*a[2,2]*a[1,3]*a[3,4]  + a[4,0]*a[0,1]*a[2,2]*a[3,3]*a[1,4]
        + a[4,0]*a[0,1]*a[3,2]*a[1,3]*a[2,4]  + a[4,0]*a[0,1]*a[3,2]*a[2,3]*a[1,4]
        + a[4,0]*a[1,1]*a[0,2]*a[2,3]*a[3,4]  + a[4,0]*a[1,1]*a[0,2]*a[3,3]*a[2,4]
        + a[4,0]*a[1,1]*a[2,2]*a[0,3]*a[3,4]  + a[4,0]*a[1,1]*a[2,2]*a[3,3]*a[0,4]
        + a[4,0]*a[1,1]*a[3,2]*a[0,3]*a[2,4]  + a[4,0]*a[1,1]*a[3,2]*a[2,3]*a[0,4]
        + a[4,0]*a[2,1]*a[0,2]*a[1,3]*a[3,4]  + a[4,0]*a[2,1]*a[0,2]*a[3,3]*a[1,4]
        + a[4,0]*a[2,1]*a[1,2]*a[0,3]*a[3,4]  + a[4,0]*a[2,1]*a[1,2]*a[3,3]*a[0,4]
        + a[4,0]*a[2,1]*a[3,2]*a[0,3]*a[1,4]  + a[4,0]*a[2,1]*a[3,2]*a[1,3]*a[0,4]
        + a[4,0]*a[3,1]*a[0,2]*a[1,3]*a[2,4]  + a[4,0]*a[3,1]*a[0,2]*a[2,3]*a[1,4]
        + a[4,0]*a[3,1]*a[1,2]*a[0,3]*a[2,4]  + a[4,0]*a[3,1]*a[1,2]*a[2,3]*a[0,4]
        + a[4,0]*a[3,1]*a[2,2]*a[0,3]*a[1,4]  + a[4,0]*a[3,1]*a[2,2]*a[1,3]*a[0,4]
```

A.2 Data File Format: `.counted`

In order to efficiently store coincidence-counting data generated by the DPC-230, we designed a custom binary file format. These files, assigned the extension `.counted`, are structured in records of three 4-byte words. The first word denotes the type of data in the record, and the following two words encode that data, as follows:

First word	Value	Meaning
MAGIC	1337	Identifies the file as being .COUNTED format.
TEMPORARY_FILE	101	Marks the file up as being temporary
STOP_METADATA	102	Marks the end of the metadata header
SCAN_TYPE	103	101: Dip/fringe, 102: Static sample, 103: Scripted scan
SCAN_NSTEPS	201	Number of steps per loop
SCAN_NLOOPS	202	Number of repeated loops in total scan
SCAN_INTEGRATION_TIME	203	Integration time per measurement, ms
SCAN_CLOSE_SHUTTER	204	Whether or not the laser shutter was closed at the end of the scan
SCAN_DONT_MOVE	205	If true, motors were disabled during the scan
SCAN_MOTOR_CONTROLLER	206	Index number of the motor controller used.
SCAN_START_POSITION	207	Motor controller position at start of scan, mm / degrees
SCAN_STOP_POSITION	208	Motor controller position at end of scan, mm / degrees
SCAN_LABEL_NBYTES	250	Length in bytes of a text label, which follows this record
Measurement data		
MOTOR_CONTROLLER_UPDATE	301	Records motor controller index and position.
SCAN_LOOP	302	Loop index
SCAN_STEP	303	Step index
INTEGRATION_STEP	304	Integration step number
STOP_INTEGRATING	305	Written when integration has finished
START_COUNT_RATES	401	Start a list of measured countrates
COUNT_RATE	402	Detection pattern as a binary string, and number of events
STOP_COUNT_RATES	403	End the list of countrates
START_PAUSE	404	Experimentalist paused the measurement
STOP_PAUSE	405	Experimentalist resumed the measurement

Now that the counting system is a little more mature, this format should really be retired in favour of a less opaque standard.

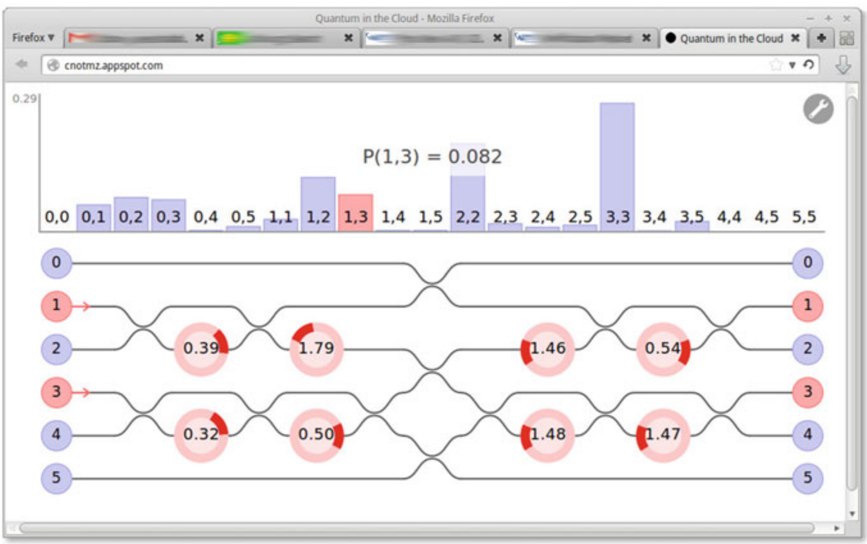

Fig. A.2 Accessible multiphoton simulation of the CNOT-MZ, running in a web browser

A.3 CNOT-MZ API

Much is made in the popular press of the potential impact and power of quantum computing, however the subject is still treated with a certain amount of trepidation, owing to the percieved difficutly of the field, creating barriers to entry for engineers and scientists from other disciplines. In an effort to make quantum computing somewhat more tangible, we built an open-access interface to the CNOT-MZ, accessible through a web browser. Users can run simulations of multiphoton experiments (Fig. A.2), either through a graphical user interface (GUI), or using an hypertext transfer protocol (HTTP) JSON application protocol interface (API). Once granted permission, they can then acquire data from the lab in real-time.

For further detail, see

https://cnotmz.appspot.com

Appendix B
Metadata

In writing my thesis, the Ph.D. theses of my colleagues and forbears (notably Jonathan Matthews, Alberto Politi, Alberto Peruzzo, Damien Bonneau, Dylan Saunders and Nathan Langford) have been an indispensable source of detailed, relevant information, clear explanation, and a model for the style and structure of a thesis.

In the hope that it might be useful to other Ph.D. students going through the same process, I include the dataset shown in Fig. B.1. This is a log of approximate word count of my thesis, recorded every time i committed a revision to my `git` repository. Three aspects of this figure are interesting: first, the striking linearity of the curve, which i absolutely expected to be a polynomial with positive second derivative. Second, one can easily identify regions of "burnout" directly after large streaks of progress: I would suggest that this stop-start mode of operation be avoided as far as possible. The last observation, I will leave as an exercise for the reader.

B.1 Dr Peter Shadbolt

Curriculum Vitae

- Web: peteshadbolt.co.uk
- Email: hello@peteshadbolt.co.uk
- Code: github.com/peteshadbolt
- Address: Controlled Quantum Dynamics, Level 12 EEE, Imperial College London

B.2 Academic

- **2015–:** PDRA at Imperial College
  - Theory of measurement-based linear-optical quantum computing
  - Solid-state sources of entangled photons (NV diamond, quantum dot)
  - Quantum algorithms for quantum chemistry

P. Shadbolt, *Complexity and Control in Quantum Photonics*,
Springer Theses, DOI 10.1007/978-3-319-21518-1

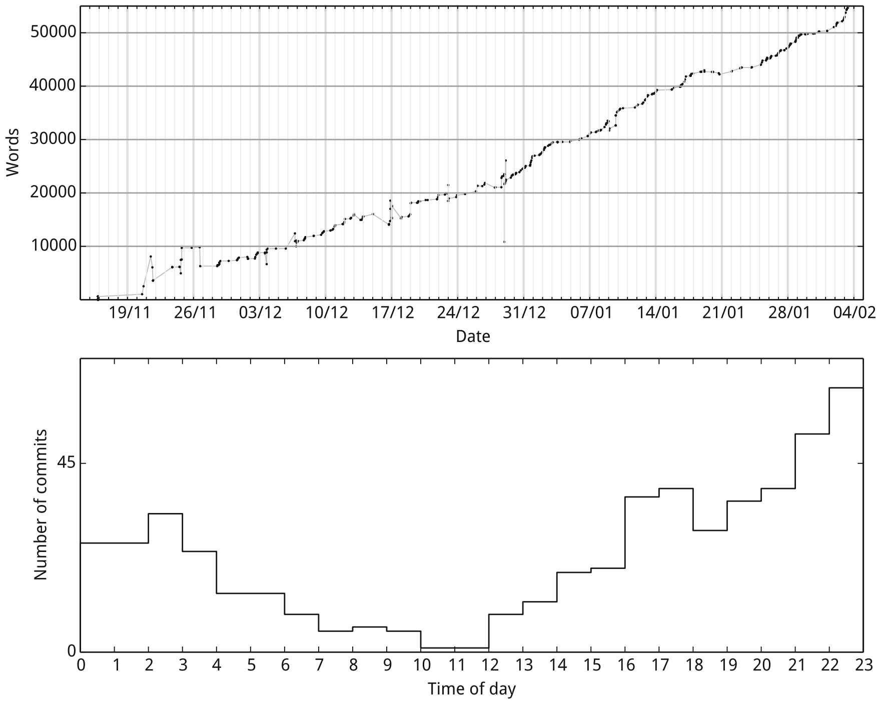

Fig. B.1 Writing a Ph.D. thesis

- **2009–2014:** Ph.D. experimental quantum photonics, University of Bristol
  - Integrated quantum photonics: Generation, manipulation and measurement of entanglement and mixture in silica-on-silicon waveguide circuits.
  - Foundations: reference-frame independent Bell inequalities, a quantum delayed choice experiment.
  - Quantum simulation of Helium Hydride on a photonic chip.
  - Quantum walks and BOSONSAMPLING, generation and measurement of 40,000-dimensional multiphoton states.
  - Construction of a time-correlated single-photon counting system using 16 avalanche-diode photon detectors.
- **2005–2009** MPhys Physics (1st Class Hons), University of Leeds
  - A novel field ionization detector for the micromaser *2008–2009, University of Leeds*
  - Studies of micromechanically exfoliated graphene *2008, University of Leeds*
  - Polymer gel electrolytes for lithium ion batteries *2007, University of Leeds*
- **1999–2005** Royal Grammar School, Buckinghamshire, England
  - 4 A Levels, 11 GCSEs and Advanced Mathematics

B.3 Publications

- P. Shadbolt, M.R. Verde, A. Peruzzo, A. Politi, A. Laing, M. Lobino, J.C.F. Matthews, J.L. O'Brien, Generating, manipulating and measuring entanglement and mixture with a reconfigurable photonic circuit. Nat. Photonics **6**, 45–49 (2012)
- A. Peruzzo, P. Shadbolt, N. Brunner, S. Popescu, J.L. OăĂŹBrien, A quantum delayed-choice experiment. Science **338**, 634–637 (2012)
- P. Shadbolt, J.C.F. Mathews, A. Laing, J.L. O'Brien, Testing foundations of quantum mechanics with photons. Nat. Phys. **10**, 278–286 (2014)
- P.J. Shadbolt, T. Vertesi, Y.C. Liang, C. Branciard, N. Brunner, J.L. O'Brien, Guaranteed violation of a Bell inequality without aligned reference frames or calibrated devices. Sci. Rep. **2**, 470 (2012)
- J. Carolan, J.D.A. Meinecke, P. Shadbolt, N.J. Russell, N. Ismail, K. Worhoff, T. Rudolph, M.G. Thompson, J.L. O'Brien, J.C.F. Matthews, A. Laing, On the experimental verification of quantum complexity in linear optics. Nat. Photonics **8**, 621–626 (2014)
- A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P.J. Love, A. Aspuru-Guzik, J.L. O'Brien, A variational eigenvalue solver on a quantum processor. Nat. Commun. **5**, 4213 (2014)
- H.W. Li, J. Wabnig, D. Bitauld, P. Shadbolt, A. Politi, A. Laing, J.L. O'Brien, A.O. Niskanen, Calibration and high fidelity measurement of a quantum photonic chip. New J. Phys. **15** (2013)
- H.W. Li, S. Przeslak, A.O. Niskanen, J.C.F. Matthews, A. Politi, P. Shadbolt, A. Laing, M. Lobino, M.G. Thompson, J.L. O'Brien, Reconfigurable controlled two-qubit operation on a quantum photonic chip. New J. Phys. **13** (2011)
- J.C.F. Matthews, X.-Qi Zhou, H. Cable, P. Shadbolt, D.J. Saunders, G.A. Durkin, G.J. Pryde, J.L. O'Brien, Practical quantum metrology. arXiv:1307.4673
- M. Gimeno-Segovia, P. Shadbolt, D.E. Browne, T. Rudolph. From three-photon GHZ states to universal ballistic quantum computation (2014). arXiv:1410.3720

B.4 Conference Talks

- Photon10, Southampton (2010) *Poster prize*,
- SPIE Optics and Optoelectronics, Prague (2011) *Invited*,
- Quantum Information Processing and Communication, Zurich (2011) *Invited*,
- Southwest Quantum Information and Technology, New Mexico (2012) *Invited*,
- CLEO 2012, San Jose (2012),
- PRACQSYS 2012, Tokyo (2012) *Invited*,
- QuAMP 2012, Oxford (2012) *Poster*,
- Bristol Cycle Festival, Bristol (2012),
- IOP Quantum technologies: taking concepts through to implementations, London (2012) *Poster*,

- Quantum Optics in the Solid State, Sheffield (2012) *Invited*,
- 13th International conference on squeezed states and uncertainty relations, Nuremberg (2013) *Invited*,
- SPIE Optics and Photonics, San Diego (2013) *Invited* $\times 2$,
- GeneExpression Systems Quantum Science Symposium, Boston (2013) *Invited*,
- Quantum Simulations, Benasque (2013) *Invited*,
- Quantum Simulations and Quantum Walks, Pisa (2013) *Invited*

B.5 Awards

- Photon10 Poster prize
- 2014 EPSRC Rising Star
- 2014 BSA Media Fellow
- Nominated 2015 Springer Theses

B.6 Outreach

- Royal Society summer exhibition *Summer 2011, London*
- Public lecture, bicycle physics *Bristol Cycle Festival 2012*
- Public lecture, nonlocality @*Bristol Science Museum 2013*
- Public lecture, quantum physics *Cafe Kino 2013*
- Web interface to a two-qubit photonic chip: http://www.cnotmz.appspot.com, >40,000 hits

B.7 Personal

Built a machine so that goldfish can play the drums and make drawings. Built a diffuse-illumination multi-touch surface for the Royal Society. Public lectures on bike physics, quantum physics, nonlocality. Black belt WTF Taekwondo. I make electronic music and films, I brew beer. I can hold a conversation in French.