UNIVERSITY OF OXFORD
SOFTWARE ENGINEERING PROGRAMME

Wolfson Building, Parks Road, Oxford OX1 3QD, UK
Tel +44(0)1865 283525 Fax +44(0)1865 283531
info@softeng.ox.ac.uk www.softeng.ox.ac.uk

Part-time postgraduate study in software engineering

Quantum Computing, QUC

7th – 11th March 2022
ASSIGNMENT

The purpose of this assignment is to test the extent to which you have achieved the learning objectives of the course. As such, your answer must be substantially your own original work. Where material has been quoted, reproduced, or co-authored, you should take care to identify the extent of that material, and the source or co-author.

Your answers to the questions on this assignment should be submitted using the Software Engineering Programme website — `www.softeng.ox.ac.uk` — following the submission guidelines. When submitting the assignment online, it is important that you formally complete all three assignment submission steps: step 1, read through the declaration; step 2, upload your files; step 3, check your files. **Please ensure your submission is anonymous: do not include your name or any other identifying information on the assignment, nor in accompanying material such as source code, nor within the file names of anything submitted.**

The deadline for submission is 12 noon on Tuesday, 26th April 2022. You are strongly encouraged to submit a version well before the deadline. You may update your submission as often as you like before the deadline, but no submissions or changes will be accepted after the deadline.

We hope to have preliminary results and comments available during the week commencing Monday, 6th June 2022. The final results and comments will be available after the subsequent examiners' meeting.

ANY QUERIES OR REQUESTS FOR CLARIFICATION REGARDING THIS ASSIGNMENT OR PROBLEMS INSTALLING SOFTWARE SHOULD, IN THE FIRST INSTANCE, BE DIRECTED TO THE PROGRAMME OFFICE WITHIN THE NEXT TWO WEEKS.

QUC Assignment (Mar 2022)

Tasks

Read arXiv:1704.05018, a seminal work about the Variational Quantum Eigensolver (VQE). Our focus is on pp.1–5, but you should also read Appendices I, II and III for additional context.

Task 1

Explain, in at most 500 words, the salient points of the paper. You should phrase your explanation in terms of concepts and notations from the QUC module (as much as possible). Feel free to use diagrams and/or sketches if you feel it helps better convey your points.

Task 2

Implement the three ansatzes from Figure 3 (see also Figure 1, for the general pattern). Use parametric controlled X rotations as entanglers, instead of the "cross-resonance gates" from the paper.

Task 3

Use VQE on an ideal simulator to find (approximate) minimum energies of the three molecules at bond distance (see Figures 2 and 3): the paper uses SPSA, but you should use COBYLA instead. The Hamiltonians for the three molecules are given by Table S2, pp.21–23. You should attempt to compute the VQE objective function using the minimum possible number of tomographic measurements, by marginalisation: you will need to modify the code from the lectures to do so, and you should briefly explain how the grouping of the Hamiltonian terms in Table S2 helps in this regard.

Some differences between your results and those of the paper are to be expected, and you should briefly comment on them.

Task 4

Implement the Hamiltonian for the Heisenberg model on a square lattice, described towards the end of p.4. Use VQE on an ideal simulator (with COBYLA) to approximately reproduce the results of Figure 4.

Some differences between your results and those of the paper are to be expected, and you should briefly comment on them.

Task 5 (extra credit)

Attempt to approximately reproduce the results of Figure S9 (plots and histograms). Some differences between your results and those of the paper are to be expected, and you should briefly comment on them.

Submission

You should submit a single Jupyter notebook with all your answers. You should save it after you've run your experiments, so I don't have to run them myself. However, it should be possible for me to re-run your notebook and (approximately) reproduce your results. For each task, you should provide a very brief commentary (2-3 lines max) on what you've done and why it works. If you cannot complete a task, you should comment on the issues that you have encountered and describe how you intended your implementation to work.

Building molecular Hamiltonians in Python

The following code snippet shows you how to quickly turn the data from Table S2 into a Python dictionary for your Hamiltonians (the example below is restricted to 2 groups for the BeH_2 molecule, but you should of course include all of them!)

```
H_BeH2 = {}
# Group 15:
s = """
XXZZXX
0.010336
IXIZXX
-0.005725
IIIZIX
0.002246
"""
a = [x for x in s.split("\n") if x]
H_BeH2.update(zip((s[::-1] for s in a[::2]), (float(x) for x in a[1::2])))
# Group 16:
s = """
YYIZXX
0.010336
IYYZXX
0.010600
"""
a = [x for x in s.split("\n") if x]
H_BeH2.update(zip((s[::-1] for s in a[::2]), (float(x) for x in a[1::2])))
# Hamiltonian being constructed:
H_BeH2
```

```
{'XXZZXX': 0.010336,
 'XXZIXI': -0.005725,
 'XIZIII': 0.002246,
 'XXZIYY': 0.010336,
 'XXZYYI': 0.0106}
```