Quantum phase estimation
-
An example from chemistry

Bertrand Marchand
PhD student @ LIX

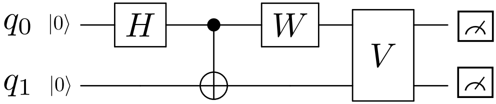
December 3, 2021

Program

1. Some context with these slides.
2. Installation problems ?
3. Your turn. Fill the “quantum_tp_pea.ipynb” according to “TP_instructions.pdf”. You will find them in Teams: Files → TP_programmation → TP_kit_phase_estimation
4. Finish it and send it[1] by **Thursday 14 January**.
5. Two important keyboard short-cuts to know: Shift+Enter and Ctrl+Enter.

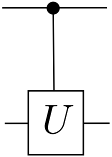
[1] the .ipynb file. Don't forget to save from time to time.

Prequisite: quantum circuits



- qubit: 2-level quantum system ($|0\rangle, |1\rangle$). quantum computer $=$ *n* qubits.
- quantum circuit $\simeq$ quantum program $=$ list of quantum instructions/gates.
- The circuit above computes: $|\psi\rangle = V \cdot (W \otimes I) \cdot CNOT_{0\to 1} \cdot (H \otimes I)|00\rangle$

Note: "controlled-gates" means:



$$= \begin{array}{cccc} \multicolumn{2}{c}{q_0 = 0} & \multicolumn{2}{c}{q_0 = 1} \\ q_1, 0 & q_1, 1 & q_1, 0 & q_1, 1 \end{array}$$
$$\left(\begin{array}{c|c} \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} & (0) \\ \hline (0) & U \end{array}\right)$$

Quantum simulation: general motivation

- Efficient numerical simulation quantum phenomena on classical machines seems hard [1].
- Therefore quantum machines might be able to perform classically-intractable computation[2]
- And in particular they should be good at simulating quantum phenomena (e.g simulating QCD, solving condensed-matter problems, computing molecular energies etc.)

—

→ But in practice, how would that work ? What **quantum circuits** would I run on my hypothetical quantum computers to solve these problems ?

[2] basic motivation behind quantum computing research

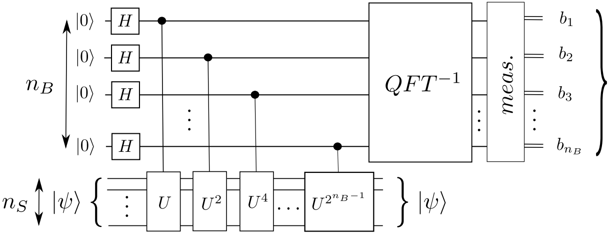
Quantum simulation: more precisely

Most of the time in the potential applications I mentioned (particle phycics, condensed matter, chemistry), the computation consists in **diagonalizing Hamiltonians** and in particular find their **ground state energy**.

You will see in these practical quantum programming sessions the two main frameworks for doing so on a quantum computer:

- **phase estimation** (this week) → the "classic guaranteed-speedup quantum algorithmics approach".
- **variational quantum algorithms** [2] (next week, in a slightly more "combinatorial optimization"-oriented context) → the "more recent, empirical, practical approach".

This week's answer: quantum phase estimation



$$\phi = 0.b_1b_2b_3\cdots b_{n_B} = \sum_k \frac{b_k}{2^k} = \frac{b_1}{2} + \frac{b_2}{4} + \cdots$$

With $|\psi\rangle$ **eigenvector** of U with **eigenvalue** $e^{-2i\pi\phi}$,
$(U|\psi\rangle = e^{-2i\pi\phi}|\psi\rangle,\ 0 \leq \phi < 1)$

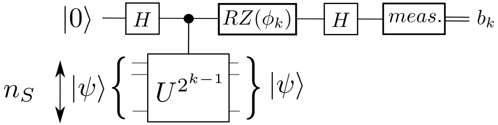
The circuit above allows to compute the n_B most significant bits of ϕ in a **binary decomposition**.

Template for *all* quantum algorithms with **exponential speedup**, both in quantum simulation and other application contexts.

Iterative quantum phase estimation: what we will use

for k in **range**(n_B, 0, -1)[3]:

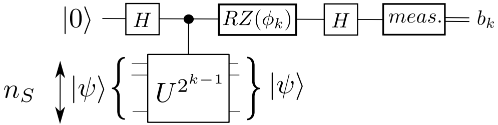
1. compute $\phi_k = 2\pi \cdot \sum_{l=k+1}^{n_B} \frac{b_l}{2^{l-k+1}}$
2. produce and execute the following circuit:



which yields b_k.

With $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ and $RZ(\phi_k) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi_k} \end{pmatrix}$

[3] Python style



Recall that $\phi = 0.b_1 \cdots b_{n_B}$ in binary decomposition,
$\phi_k = 2\pi \cdot \sum_{l=k+1}^{n_B} \frac{b_l}{2^{l-k+1}} = 0.0b_{k+1} \cdots b_{n_B}$ and $U|\psi\rangle = e^{-2i\pi\phi}|\psi\rangle$.

Evolution of the system:

$$|0\rangle|\psi\rangle \underset{H}{\rightarrow} \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)|\psi\rangle = \left(\frac{|0\rangle|\psi\rangle+|1\rangle|\psi\rangle}{\sqrt{2}}\right)$$

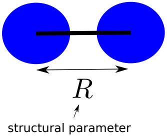
$$\underset{\bullet-U}{\hookrightarrow} \left(\frac{|0\rangle|\psi\rangle+e^{-2i\pi\cdot 2^{k-1}\phi}|1\rangle|\psi\rangle}{\sqrt{2}}\right) = \left(\frac{|0\rangle+e^{-2i\pi\cdot 0.b_k\cdots b_{n_B}}|1\rangle}{\sqrt{2}}\right)|\psi\rangle$$

$$\underset{RZ(\phi_k)}{\hookrightarrow} \left(\frac{|0\rangle+e^{-2i\pi\cdot 0.b_k}|1\rangle}{\sqrt{2}}\right)|\psi\rangle = \left(\frac{|0\rangle+(-1)^{b_k}|1\rangle}{\sqrt{2}}\right)|\psi\rangle \underset{H}{\rightarrow} |b_k\rangle|\psi\rangle$$

Our example: molecular ground state energy calculation

General problem: Given $\mathcal{H}$, electronic structure Hamiltonian of a molecule, we want to compute its ground state energy.

Actually, usually $\mathcal{H}$ depends on some structural/geometric parameters. The purpose is then to compute a *ground state energy landscape* as a function of these parameters. May provide insight on reaction mechanisms [4].



→

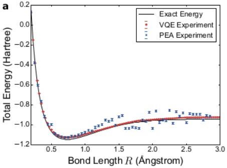
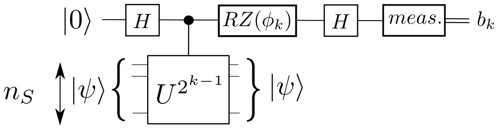


FIG. 3. **Computed H_2 energy curve and errors.** (a) Ene

Our instance: H_2 and its dissociation curve. We will use data from [4], where they solve it on an actual small quantum chip.

[4] Potential futuristic applications: CO_2 capture [3], H_2O cracking...



We will therefore work with:

- $U = e^{-i\mathcal{H}dt}$, for some value of dt to be chosen.
- $|\psi\rangle$ the ground state of H, whose energy E we want to compute.
- we then have $U|\psi\rangle = e^{-iEdt}|\psi\rangle = e^{-2i\pi\frac{Edt}{2\pi}}$

Problem 1: the algorithm only allows us to compute $\phi = \frac{Edt}{2\pi}$ mod $[1]$ $(\in [0,1])$

Following [5], we will use an *interval guess* $[E_{min}, E_{max}]$ for E. Then $\mathcal{H} \leftarrow \mathcal{H} - E_{min} \cdot I$ ensures $E > 0$ and choosing $dt = \frac{2\pi}{E_{max}}$ ensures $\frac{Edt}{2\pi} < 1$

Other problems

Problem 2 In general we do not have access to the ground state.
→use an **ansatz**[5] with theoretical good overlap with ground state. As suggested in [4], we will use here the Hartree-Fock approxiamtion to our problem, namely $|10\rangle = X \otimes I|00\rangle$

Problem 3 How do we implement $U^{2^k} = e^{-i\mathcal{H}2^k dt}$?
→ usually $\mathcal{H}$ comes as a sum of simpler terms $\sum_l \mathcal{H}_l$ (e.g Pauli products) for which $e^{-i\mathcal{H}_l dt}$ is simple to implement.
→ and use **Trotter's formula** [5]

$$e^{-i\left(\sum_l \mathcal{H}_l\right)dt} \simeq \left(\prod_l e^{-\frac{i\mathcal{H}_l dt}{p}}\right)^p$$

with p integer, as large as possible.

[5] easy-to-prepare approximation

- We will reproduce the curves from [4].
- In this example:
  - $n_S = 2$
  - $H(R) = g_0 I + g_1 ZI + g_2 IZ + g_3 ZZ + g_4 YY + g_5 XX$[6]
  - The coefficients are stored in hamiltonian_data.json
- The purpose is to reproduce

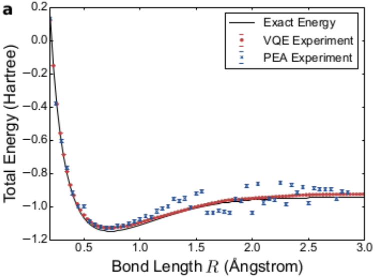


FIG. 3. **Computed H_2 energy curve and errors.** (a) Ene

[6] This Hamiltonian is the output of (1) a conversion from an *electronic structure Hamiltonian* to a *spin Hamiltonian* expressed with Pauli matrices and (2) simplifications leveraging theoretical chemistry knowledge

The programming framework we will use: myqlm

What you need to know:

- quantum circuits are first written as **Program** and then exported as **circuit**.
- A **job** is created from a circuit and sent to a **qpu** (a simulator)
- **AbstractGates** allow to define custom parameterized gates. They will come handy in programming Hamiltonian simulation.
- **QRoutines** allow to program arbitrary parametrized sub-circuits, which are then pluggable anywhere into a larger circuit.
- see the minimal notebook and https://myqlm.github.io/ for more details

Richard P Feynman.
Simulating physics with computers.
Int. J. Theor. Phys, 21(6/7), 1982.

M Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al.
Variational quantum algorithms.
arXiv preprint arXiv:2012.09265, 2020.

Vera von Burg, Guang Hao Low, Thomas Häner, Damian S Steiger, Markus Reiher, Martin Roetteler, and Matthias Troyer.
Quantum computing enhanced computational catalysis.
arXiv preprint arXiv:2007.14460, 2020.

Peter JJ O'Malley, Ryan Babbush, Ian D Kivlichan, Jonathan Romero, Jarrod R McClean, Rami Barends, Julian Kelly, Pedram Roushan, Andrew Tranter, Nan Ding, et al.
Scalable quantum simulation of molecular energies.

Physical Review X, 6(3):031007, 2016.

James D Whitfield, Jacob Biamonte, and Alán Aspuru-Guzik.
Simulation of electronic structure hamiltonians using quantum computers.
Molecular Physics, 109(5):735–750, 2011.