

**Physics at the Perimeter Institute:  
Topics in Theoretical Physics  
&  
Quantum Simulation Research  
- A Quick Guide -**

Huan Q. Bui

Colby College

PHYSICS & MATHEMATICS  
Statistics

Class of 2021

May 23, 2020

## **Preface**

Greetings,

This guide is my notes from Perimeter Institute of Theoretical Physics Summer School, 2020. Topics include quantum information, thermodynamics, numerical methods, condensed matter physics, path integrals, and symmetries.

The text also includes reading notes for my research project at Perimeter on quantum simulation.

Enjoy!

# Contents

Preface . . . . .	2
<b>1 Quantum Information &amp; Thermodynamics</b>	<b>7</b>
1.1 Foundations of Quantum Statistical Mechanics - Entanglement .	8
1.2 Resource Theories and Quantum Information . . . . .	9
1.3 Quantum Thermal Operations . . . . .	10
1.4 Fluctuation Theorems and Quantum Information . . . . .	11
1.5 Quantum Thermal Machines . . . . .	12
<b>2 Numerical Methods &amp; Condensed Matter Physics</b>	<b>13</b>
2.1 L1: Introduction to many-particle states and operators . . . . .	14
2.1.1 L1: Introduction to Ising and XXZ models . . . . .	15
2.1.2 L1: Programming basics . . . . .	16
2.1.3 L1 : Finding expectation values . . . . .	17
2.2 L2: Exact diagonalization part 1 . . . . .	18
2.3 L2: Representing models . . . . .	19
2.4 L2: Finding eigenstates . . . . .	20
2.5 L2: Energy gaps . . . . .	21
2.6 L2: Phase transitions . . . . .	22
2.7 L3: Exact diagonalization part 2 . . . . .	23
2.8 L3: Limitations of the method . . . . .	24
2.9 L3: Using symmetries . . . . .	25
2.10 L3: Dynamics . . . . .	26
2.11 L4: Matrix product states part 1 . . . . .	27
2.12 L4: Entanglement and the singular value decomposition . . . . .	28
2.13 L4: What is a matrix product state and why is it useful . . . . .	29
2.14 L5: Matrix product states part 2 . . . . .	30
2.15 L5: Algorithms for finding ground states using matrix product states: iTEBD . . . . .	31
2.16 L5: Algorithms for finding ground states using matrix product states: DMRG . . . . .	32
<b>3 Path Integrals</b>	<b>33</b>
3.1 Introduction to path integrals and the semi-classical limit . . . . .	34
3.2 Propagator in real and imaginary time . . . . .	35

3.3	Perturbation theory . . . . .	36
3.4	Non-perturbative physics and quantum tunneling . . . . .	37
3.5	Topology and path integrals . . . . .	38
<b>4</b>	<b>Symmetries</b>	<b>39</b>
4.1	Review: Equivalence Classes . . . . .	40
4.1.1	Equivalence relations . . . . .	40
4.1.2	Equivalence Classes . . . . .	40
4.2	L1: Definition of symmetry . . . . .	43
4.3	L1: Elements of group theory . . . . .	44
4.4	L1: Examples: Symmetry & Group Theory . . . . .	45
4.5	L2: Continuous and discrete symmetries . . . . .	46
4.6	L2: Infinitesimal symmetries . . . . .	47
4.7	L2: Noether's theorem . . . . .	48
4.8	L3: Lie groups and Lie algebras . . . . .	49
4.9	L4: Symmetries in quantum mechanics . . . . .	50
4.10	L5: Representation theory . . . . .	51
<b>5</b>	<b>Research Project: Quantum Simulation</b>	<b>53</b>
5.1	Review: The density operator . . . . .	54
5.1.1	Ensembles of Quantum States . . . . .	54
5.1.2	General Properties of the Density Operator . . . . .	55
5.1.3	The Reduced Density Operator . . . . .	57
5.2	Quantum Computational Complexity: An Introduction . . . . .	58
5.2.1	Introduction . . . . .	58
5.3	Quantum Simulation: A Primer . . . . .	59
5.3.1	Simulation in action . . . . .	59
5.3.2	The quantum simulation algorithm . . . . .	60
5.3.3	The algorithm . . . . .	62
5.3.4	An illustrative example . . . . .	62
5.3.5	Perspectives on quantum simulation . . . . .	63
5.4	Problems in Quantum Simulations and Quantum Algorithms . . . . .	65
5.4.1	Quantum Simulation: An Overview . . . . .	65
5.4.2	Simulation challenges in molecular and materials science . . . . .	67
5.4.3	Challenges for quantum algorithms in quantum simulation . . . . .	69
5.5	Some Models in Quantum Statistical & Condensed-Matter Physics . . . . .	77
5.5.1	The Hubbard Model . . . . .	77
5.5.2	The Ising Model . . . . .	77
5.6	Quantum Phase Estimation (QPE) . . . . .	78
5.6.1	Performance and Requirements . . . . .	80
5.6.2	The algorithm . . . . .	82
5.7	Quantum Variational Eigensolver (QVE) . . . . .	84
5.7.1	Quantum Expectation Estimation (QEE) . . . . .	84
5.7.2	Quantum Variational Eigensolver (QVE) . . . . .	87
5.8	Quantum Approximate Optimization Algorithm (QAOA) . . . . .	89
5.9	Quantum Variational Factorization (QVF) . . . . .	90

5.10 Variational Thermal Quantum Simulation via Thermofield Double States . . . . .	91
5.10.1 Thermofield Double States (TFD) . . . . .	91
5.11 Generation of TFD and Critical Ground States with a Quantum Computer . . . . .	92
5.12 Efficient variational simulation of non-trivial quantum states . .	93



## Part 1

# Quantum Information & Thermodynamics

The aim of this course is to understand the thermodynamics of quantum systems and in the process to learn some fundamental tools in Quantum Information. We will focus on the topics of foundations of quantum statistical mechanics, resource theories, entanglement, fluctuation theorems, and quantum machines.

## **1.1 Foundations of Quantum Statistical Mechanics - Entanglement**



## 1.2 Resource Theories and Quantum Information

### **1.3 Quantum Thermal Operations**

## 1.4 Fluctuation Theorems and Quantum Information

## **1.5 Quantum Thermal Machines**

## Part 2

# Numerical Methods & Condensed Matter Physics

This course has two main goals: (1) to introduce some key models from condensed matter physics; and (2) to introduce some numerical approaches to studying these (and other) models. As a precursor to these objectives, we will carefully understand many-body states and operators from the perspective of condensed matter theory. (However, I will cover only spin models. We will not discuss or use second quantization.)

Once this background is established, we will study the method of exact diagonalization and write simple python programs to find ground states, correlation functions, energy gaps, and other properties of the transverse-field Ising model and XXZ model. We will also discuss the computational limitations of exact diagonalization. Finally, I will introduce the concept of matrix product states, and we will see how these can be used with algorithms such as the density matrix renormalization group (DMRG) to study ground state properties for much larger systems than can be studied with exact diagonalization.

## **2.1 L1: Introduction to many-particle states and operators**

2.1. *L1: INTRODUCTION TO MANY-PARTICLE STATES AND OPERATORS*15

**2.1.1 L1: Introduction to Ising and XXZ models**

### **2.1.2 L1: Programming basics**



2.1. *L1: INTRODUCTION TO MANY-PARTICLE STATES AND OPERATORS*17

**2.1.3 L1 : Finding expectation values**

## **2.2 L2: Exact diagonalization part 1**

## **2.3 L2: Representing models**

## **2.4 L2: Finding eigenstates**

## 2.5 L2: Energy gaps

## **2.6 L2: Phase transitions**

## **2.7 L3: Exact diagonalization part 2**

## **2.8 L3: Limitations of the method**



## **2.9 L3: Using symmetries**

## **2.10 L3: Dynamics**

## **2.11 L4: Matrix product states part 1**

## **2.12 L4: Entanglement and the singular value decomposition**

2.13. L4: WHAT IS A MATRIX PRODUCT STATE AND WHY IS IT USEFUL29

**2.13 L4: What is a matrix product state and why is it useful**

## **2.14 L5: Matrix product states part 2**

**2.15 L5: Algorithms for finding ground states  
using matrix product states: iTEBD**

**2.16 L5: Algorithms for finding ground states  
using matrix product states: DMRG**



## Part 3

# Path Integrals

The goal of this course is to introduce the path integral formulation of quantum mechanics and a few of its applications. We will begin by motivating the path integral formulation and explaining its connections to other formulations of quantum mechanics and its relation to classical mechanics. We will then explore some applications of path integrals.

### 3.1 Introduction to path integrals and the semi-classical limit

## 3.2 Propagator in real and imaginary time

### 3.3 Perturbation theory

### 3.4 Non-perturbative physics and quantum tunneling

### 3.5 Topology and path integrals

## Part 4

# Symmetries

The aim of this course is to explore some of the many ways in which symmetries play a role in physics. We'll start with an overview of the concept of symmetries and their description in the language of group theory. We will then discuss continuous symmetries and infinitesimal symmetries, their fundamental role in Noether's theorem, and their formalization in terms of Lie groups and Lie algebras. In the last part of the course we will focus on symmetries in quantum theory and introduce representations of (Lie) groups and Lie algebras.

## 4.1 Review: Equivalence Classes

### 4.1.1 Equivalence relations

An equivalence relation is a special kind of binary relation. A (homogeneous) binary relation on a set  $X$  is a way of encoding a relationship between two elements of  $X$ . Given a relation  $R$  we use the notation  $xRy$  to mean “ $x$  is related to  $y$ .”

Examples of relations are  $=, <, >, \leq, \geq$  on  $\mathbb{R}$ .

**Definition 4.1.1** (Homogeneous binary relation). A homogeneous binary relation on a set  $X$  is a set  $R \subset X \times X$  of pairs of elements of  $X$ . When  $(x, y) \in R$  we say that  $x$  is related to  $y$ , which we also indicate with the less cumbersome notation  $xRy$ .

We can now define equivalence relations.

**Definition 4.1.2** (Equivalence relation). An equivalence relation on a set  $X$  is a homogeneous binary relation  $R$  that satisfies the following properties:

- Reflexive:  $xRx \forall x \in X$ .
- Symmetric:  $xRy \implies yRx$ .
- Transitive:  $xRy, yRz \implies xRz$ .

These properties are chosen as they encode the fundamental features of the notion of things being “equivalent.”

Equivalence relations are often denoted by the symbol  $\sim$ . In this case we read  $x \sim y$  as “ $x$  is equivalent to  $y$ .”

### 4.1.2 Equivalence Classes

Now that we know what an equivalence relation is, we can define the extremely useful concept of equivalence class.

**Definition 4.1.3.** Let  $\sim$  be an equivalence relation on  $X$ . The equivalence class of  $x \in X$  is the set

$$[x]_{\sim} = \{y \in X \mid y \sim x\} \quad (4.1)$$

of all the elements in  $X$  equivalent to  $x$ . We often drop the subscript and just write  $[x]$  if there is no risk of confusion. Note that

$$[x]_{\sim} = [y]_{\sim}, \quad \forall y \in [x]_{\sim}, \quad (4.2)$$

that is we can label the equivalence class using any of its elements. The element that we choose as the label is called the representative of the equivalence class.



Equivalence classes are essentially buckets in which we put elements of  $X$  that are equivalent to each other. Inequivalent elements end up in different equivalence classes. One way of thinking about this is that an equivalence class is a way to encode as a single object the properties common to its elements, which are otherwise indistinguishable.

**Example 4.1.1.** An example of equivalence relation appears in modular arithmetic. Let's consider the simplest case of the integers modulo 2. We define an equivalence relation  $\sim$  on  $\mathbb{Z}$  by

$$x \sim y \implies x - y = 2k, k \in \mathbb{Z}, \quad (4.3)$$

that is  $x, y$  are equivalent if they differ by a multiple of two. If you want, you can check that this is indeed an equivalence relation as an exercise. There are only two distinct equivalence classes, which are the respectively the sets of even and odd integers. The sets of integers modulo 2 is defined as

$$\mathbb{Z}_2 = \{[0], [1]\}. \quad (4.4)$$

Although this is a set of sets, we really want to pretend that it's a set of numbers. To do that, we define a way to add elements in  $\mathbb{Z}_2$ . The obvious choice is

$$[a] + [b] = [a + b], \quad (4.5)$$

but at this stage this expression is purely formal: since we are specifying this operation through the sum of the representatives of each class, we need to make sure that the definition is actually independent of the choice of representatives (since they should be equivalent). This happens all the time when dealing with equivalence classes! In this case we are fine, because of the rules of adding odd and even numbers.

**Example 4.1.2.** Here's an example from quantum physics. Let  $\mathcal{H}$  be the Hilbert space of a quantum theory. While extremely useful, the states  $|\psi\rangle \in \mathcal{H}$  are not *physical*, where physical is meant as something that can be measured. The physical quantities in a quantum theory are the expectation values of the observables in a given state, that is

$$\langle A \rangle_\psi = \frac{\langle \psi | A | \psi \rangle}{\langle \psi | \psi \rangle}. \quad (4.6)$$

Now consider another state  $|\phi\rangle = \lambda\psi$ , with  $\lambda \in \mathbb{C} \setminus \{0\}$ . For any observable  $A$  we have

$$\langle A \rangle_\phi = \frac{|\lambda|^2 \langle \psi | A | \psi \rangle}{|\lambda|^2 \langle \psi | \psi \rangle} = \langle A \rangle_\psi. \quad (4.7)$$

This means that the states  $|\psi\rangle$  and  $|\phi\rangle$  are physically indistinguishable. It is then natural to consider them equivalent, and define the equivalence relation

$$|\psi\rangle \sim |\phi\rangle \iff |\phi\rangle = \lambda |\psi\rangle, \quad \lambda \in \mathbb{C} \setminus \{0\}. \quad (4.8)$$

The set

$$P(\mathcal{H}) = \{[|\psi\rangle] \mid |\psi\rangle \in \mathcal{H} \setminus \{0\}\} \quad (4.9)$$

is called the *projective Hilbert space* associated to  $\mathcal{H}$ , and can be interpreted as the set of all distinguishable quantum states. The zero vector was removed as it does not describe any physical state (expectation values don't make sense for the zero vector).

In a sense  $P(\mathcal{H})$  is the true set of states of a quantum theory. The reason why we work with  $\mathcal{H}$  instead is that the latter is a vector space, which comes with a lot of nice properties and allows for an easy description of the superposition principle.

## 4.2 L1: Definition of symmetry

### 4.3 L1: Elements of group theory

## 4.4 L1: Examples: Symmetry & Group Theory

## 4.5 L2; Continuous and discrete symmetries

## 4.6 L2: Infinitesimal symmetries

## 4.7 L2: Noether's theorem



## 4.8 L3: Lie groups and Lie algebras

## 4.9 L4: Symmetries in quantum mechanics

## 4.10 L5: Representation theory



## Part 5

# Research Project: Quantum Simulation

### Quantum many-body physics on quantum hardware

Project description: Recently, there have been significant advances in several quantum computing platforms, including superconducting qubits and trapped ions. At this point, nontrivial quantum operations can be implemented on tens of qubits, and the frontier continues to expand. This project will explore how these emerging technologies can assist the realization and understanding of complex many-body quantum systems, regarding both static aspects such as ground state properties and dynamical aspects such as quantum chaos and thermalization. What new physics can we learn from near-term quantum computers? The ideal outcome is an interesting and realistic proposal that can be carried out in one of the quantum platforms (see for example the interplay between theoretical proposals and experiment).

## 5.1 Review: The density operator

We often use the language of state vectors in quantum mechanics. An alternative language is that of *density matrices/density operators*. We will use this language extensively in quantum information/quantum computation. First, we will introduce the formulation. Second, we look at some properties of the density operator. Finally, we look at an application where the density operator really shines – as a tool for describing *individual subsystems* of a composite quantum system.

### 5.1.1 Ensembles of Quantum States

The density operator language provides a convenient means for describing quantum systems whose state is not completely known. Suppose a quantum system is in one of the states  $|\psi_i\rangle$  where  $i$  is an index, with respective probabilities  $p_i$ . We call  $\{p_i, |\psi_i\rangle\}$  an *ensemble of pure states*.

**Definition 5.1.1** (Density Operator).

$$\rho \equiv \sum_i p_i |\psi_i\rangle \langle \psi_i| \quad (5.1)$$

Suppose we let a unitary  $\mathcal{U}$  act on a closed system. If the system was initially in the state  $|\psi_i\rangle$  with probability  $p_i$  then we get  $\mathcal{U}|\psi_i\rangle$  with probability  $p_i$ . The density operator evolves as

$$\rho = \sum_i p_i |\psi_i\rangle \langle \psi_i| \xrightarrow{\mathcal{U}} \sum_i p_i \mathcal{U}|\psi_i\rangle \langle \psi_i| \mathcal{U}^\dagger = \mathcal{U} \rho \mathcal{U}^\dagger. \quad (5.2)$$

Suppose we have a measurement gate  $\mathcal{M}_m$ . If the initial state is  $|\psi_i\rangle$  then the probability of getting result  $m$  is

$$p(m|i) = \langle \psi_i | \mathcal{M}_m^\dagger \mathcal{M}_m | \psi_i \rangle = \text{tr} (\mathcal{M}_m^\dagger \mathcal{M}_m \langle \psi_i | \psi_i \rangle) \quad (5.3)$$

where we have used the identity:

$$\text{tr}(A |\psi\rangle \langle \psi|) = \sum_i \langle i | A | \psi \rangle \langle \psi | i \rangle = \langle \psi | A | \psi \rangle. \quad (5.4)$$

With this, the total probability of measuring  $m$  is

$$p(m) = \sum_i p(m|i) p_i = \sum_i p_i \text{tr} (\mathcal{M}_m^\dagger \mathcal{M}_m |\psi_i\rangle \langle \psi_i|) = \text{tr} (\mathcal{M}_m^\dagger \mathcal{M}_m \rho). \quad (5.5)$$

The state after the measurement is thus

$$|\psi_i^m\rangle = \frac{\mathcal{M}_m |\psi_i\rangle}{\sqrt{\langle \psi_i | \mathcal{M}_m^\dagger \mathcal{M}_m | \psi_i \rangle}}. \quad (5.6)$$

The corresponding density operator for this state is thus

$$\rho_m = \sum_i p(i|m) |\psi_i^m\rangle \langle \psi_i^m| = \sum_i p(i|m) \frac{\mathcal{M}_m |\psi_i\rangle \langle \psi_i| \mathcal{M}_m^\dagger}{\langle \psi_i| \mathcal{M}_m^\dagger \mathcal{M}_m |\psi_i\rangle}. \quad (5.7)$$

From probability theory we have  $p(i|m) = p(m|i)p_i/p(m)$ , so:

$$\rho_m = \sum_i p_i \frac{\mathcal{M}_m |\psi_i\rangle \langle \psi_i| \mathcal{M}_m^\dagger}{\text{tr}(\mathcal{M}_m^\dagger \mathcal{M}_m \rho)} = \frac{\mathcal{M}_m \rho \mathcal{M}_m^\dagger}{\text{tr}(\mathcal{M}_m^\dagger \mathcal{M}_m \rho)}. \quad (5.8)$$

Finally, suppose we have a quantum system in state  $\rho_i$  with probability  $p_i$ . The system might be described by the density matrix

$$\rho = \sum_i p_i \rho_i. \quad (5.9)$$

Here's why:

$$\rho = \sum_{i,j} p_i p_{ij} |\psi_{ij}\rangle \langle \psi_{ij}| = \sum_i p_i \rho_i \quad (5.10)$$

where we have used the definition

$$\rho_i = \sum_j p_{ij} |\psi_{ij}\rangle \langle \psi_{ij}|. \quad (5.11)$$

We call  $\rho$  the *mixture* of the states  $\rho_i$  with probabilities  $p_i$ . This concept of mixture comes up repeatedly in the analysis of problems like quantum noise, where the effect of the noise is to introduce ignorance into our knowledge of the quantum state. A simple example is provided by the measurement scenario. Imagine that for some reason our record of the result  $m$  of the measurement was lost. We would have a quantum system in the state  $\rho_m$  with probability  $p(m)$ , but would no longer know the actual value of  $m$ . The state of such a quantum system would therefore be described by the density operator

$$\rho = \sum_m p(m) \rho_m = \sum_m \text{tr}(\mathcal{M}_m^\dagger \mathcal{M}_m \rho) \frac{\mathcal{M}_m \rho \mathcal{M}_m^\dagger}{\text{tr}(\mathcal{M}_m^\dagger \mathcal{M}_m \rho)} = \sum_m \mathcal{M}_m \rho \mathcal{M}_m^\dagger. \quad (5.12)$$

### 5.1.2 General Properties of the Density Operator

**Theorem 5.1.1** (Characterization of density operators). An operator  $\rho$  is the density operator to some ensemble  $\{p_i, |\psi_i\rangle\}$  if and only if it satisfies the conditions

- **Trace condition:**  $\text{tr}(\rho) = 1$ .
- **Positivity:**  $\rho$  is a positive operator, i.e.,  $\langle \varphi | \rho | \varphi \rangle \geq 0 \forall \varphi$ .

With this definition we can reformulate the postulates of quantum mechanics in the language of density operators as

- **Postulate 1:** Associated to any isolated physical system is a Hilbert space known as the *state space* of the system. The system is completely described by its *density operator*, which is a positive operator  $\rho$  with trace one, acting on the state space of the system. If a quantum system is in the state  $\rho_i$  with probability  $p_i$  then the density operator for the system is  $\sum_i p_i \rho_i$ .
- **Postulate 2:** The evolution of a *closed* quantum system is described by a *unitary transformation*:

$$\rho' = \mathcal{U} \rho \mathcal{U}^\dagger. \quad (5.13)$$

- **Postulate 3:** Measurements are described by a collection  $\{\mathcal{M}_m\}$  of *measurement operators*. The index  $m$  refers to the measurement outcomes that may occur. If the state of the quantum system is  $\rho$  immediately before the measurement then the probability that result  $m$  occurs is given by

$$p(m) = \text{tr}(\mathcal{M}_m^\dagger \mathcal{M}_m \rho_m) \quad (5.14)$$

and the state of the system after the measurement is

$$\frac{\mathcal{M}_m \rho \mathcal{M}_m^\dagger}{\text{tr}(\mathcal{M}_m^\dagger \mathcal{M}_m \rho)}. \quad (5.15)$$

The measurement operators satisfy the completeness equation

$$\sum_m \mathcal{M}_m^\dagger \mathcal{M}_m = \mathcal{I}. \quad (5.16)$$

- **Postulate 4:** The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. E.g. the joint state of the total system can be written as  $\rho_1 \otimes \cdots \otimes \rho_n$ .

**Theorem 5.1.2** (Pure states).  $\text{tr}(\rho^2) \leq 1$ . Equality occurs if and only if  $\rho$  is a pure state.

**Theorem 5.1.3** (Unitary freedom in the ensemble for density matrices). The sets  $|\tilde{\varphi}_i\rangle$  and  $|\tilde{\psi}_j\rangle$  generate the same density matrix if and only if they are unitarily similar, i.e.,

$$|\tilde{\varphi}_i\rangle = \sum_j u_{ij} |\tilde{\psi}_j\rangle \quad (5.17)$$

where the  $u_{ij}$  are entries of a unitary  $\mathcal{U}$ .



### 5.1.3 The Reduced Density Operator

The deepest application of the density operator is as a descriptive tool for *subsystems* of a composite quantum system. Such a description is given by the *reduced density operator*.

Suppose we have physical systems  $A$  and  $B$  whose state is described by the density operator  $\rho^{AB}$ . The reduced density operator for  $A$  is then given by

$$\rho^A = \text{tr}_B (\rho^{AB}) \quad (5.18)$$

where  $\text{tr}_B$  is the *partial trace* over system  $B$ , defined by

$$\text{tr}_B (|a_1\rangle \langle a_2| \otimes |b_1\rangle \langle b_2|) \equiv |a_1\rangle \langle a_2| \text{tr}(|b_1\rangle \langle b_2|) = |a_1\rangle \langle a_2| \langle b_2|b_1\rangle \quad (5.19)$$

where  $|a_1\rangle, |a_2\rangle$  are any two vectors in the state space of  $A$ , and  $|b_1\rangle, |b_2\rangle$  are any two vectors in the state space of  $B$ .

**Example 5.1.1.** To understand this construction, consider an example where  $\rho^{AB} = \rho \otimes \sigma$  where  $\rho, \sigma$  describe  $A, B$  respectively. Then

$$\rho^A = \text{tr}_B (\rho \otimes \sigma) = \rho \text{tr}(\sigma) = \rho \quad (5.20)$$

as expected.

## 5.2 Quantum Computational Complexity: An Introduction

In this section we look at quantum computational complexity, as introduced in this [paper](#) by John Watrous at the IQC, the University of Waterloo.

### 5.2.1 Introduction

- P
- NP
- BPP
- PP
- MA
- AM
- SZK
- PSPACE
- EXP
- NEXP
- PL
- NC

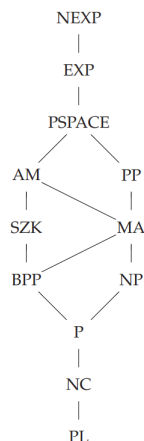


Figure 1: A diagram illustrating known inclusions among most of the classical complexity classes discussed in this paper. Lines indicate containments going upward; for example, AM is contained in PSPACE.

## 5.3 Quantum Simulation: A Primer

Here's what Feynman said in 1982:

*“Can physics be simulated by a universal computer? [...] the physical world is quantum mechanical, and therefore the proper problem is the simulation of quantum physics [...] the full description of quantum mechanics for a large system with  $R$  particles [...] has too many variables, it cannot be simulated with a normal computer with a number of elements proportional to  $R$  [...] but it can be simulated with ] quantum computer elements. [...] Can a quantum system be probabilistically simulated by a classical (probabilistic, I'd assume) universal computer? [...] If you take the computer to be the classical kind I've described so far [...] the answer is certainly, No!”*

### 5.3.1 Simulation in action

The heart of simulation is the solution of differential equations which capture the physical laws governing the dynamical behavior of a system. The goal is generally: given an initial state of the system, what is the state at some other time and/or position? Solutions are usually obtained by *approximating* the state with a digital representation, then *discretizing* the differential equation in space and time such that an iterative application of a procedure carries the state from the initial to the final conditions.

The error in this procedure is *bounded*, and known not to grow faster than some small power of the number of iterations. Furthermore, not all dynamical systems can be simulated efficiently: generally, only those systems which can be

described efficiently can be simulated efficiently.

Simulating quantum systems using classical computers is usually inefficient. The dynamical behavior of many simple quantum systems is governed the SE:

$$i\hbar\partial_t |\psi\rangle = \mathcal{H} |\psi\rangle. \quad (5.21)$$

We will set  $\hbar = 1$ . For real particles in space, we often have

$$i\partial_t\psi(x) = \left[ -\frac{1}{2m}\partial_x^2 + V(x) \right] \psi(x). \quad (5.22)$$

The key challenge in simulating quantum systems is the *exponential* number of differential equations which must be solved. For 1 qubit evolving according to the SE, a system of 2 differential equations must be solved; for 2 qubits, 4 equations; and for  $n$  qubits,  $2^n$  equations. Sometimes, insightful approximations can be made which reduce the effective number of equations involved, but there are many physically interesting quantum systems for which no such approximations are known.

There are many important quantum systems for which classical simulation is intractable. For example:

$$\text{Hubbard model: } \mathcal{H} = \sum_{k=1}^n V_0 n_{k\uparrow} n_{k\downarrow} + \sum_{k,j \text{ neighbors}, \sigma} t_0 c_{k\sigma}^* c_{j\sigma}. \quad (5.23)$$

$$\text{Ising mode: } \mathcal{H} = \sum_{k=1}^n \vec{\sigma}_k \cdot \vec{\sigma}_{k+1}. \quad (5.24)$$

Quantum computers can efficiently simulate quantum systems for which there is no known efficient classical simulation.

### 5.3.2 The quantum simulation algorithm

The quantum simulation is concerned with the solution of  $i\partial_t |\psi\rangle = \mathcal{H} |\psi\rangle$ , which, for a time-independent  $\mathcal{H}$ , is just

$$|\psi(t)\rangle = e^{-i\mathcal{H}t} |\psi(0)\rangle. \quad (5.25)$$

$\mathcal{H}$  is in general difficult to exponentiate, but a good beginning is the first order solution

$$|(\psi(t + \Delta t))\rangle \approx (\mathcal{I} - i\mathcal{H}\Delta t) |\psi(t)\rangle. \quad (5.26)$$

This is tractable, but often is not very satisfactory.

Efficient approximations to higher order is possible for many *classes* of Hamiltonians. For example, in most physical systems, the Hamiltonian can be written as a sum over many *local* interactions. For a system of  $n$  particles:

$$\mathcal{H} = \sum_{k=1}^L \mathcal{H}_k \sim \bigoplus_{k=1}^L \mathcal{H}_k \quad (5.27)$$

where each  $\mathcal{H}_k$  acts on at most a constant  $c$  number of systems, and  $L$  is a polynomial in  $n$ . For instance, the terms  $\mathcal{H}_k$  are often just 2-body interactions such as  $X_i X_j$  and 1-body Hamiltonians such as  $X_i$ . Note that both the Hubbard and Ising models have Hamiltonians of this form. There are sometimes additional global symmetry constraints such as particle statistics.

Note that although  $e^{-i\mathcal{H}t}$  is difficult to compute,  $e^{-i\mathcal{H}_k t}$  acts on a much smaller subsystem, and is straightforward to approximate using quantum circuits. However, because  $[\mathcal{H}_j, \mathcal{H}_k] \neq 0$  in general (obviously, they don't commute in general),  $e^{-i\mathcal{H}t} \neq \prod_k e^{-i\mathcal{H}_k t}$ . So, we might want to ask how  $e^{-i\mathcal{H}_k t}$  be useful in constructing  $e^{-i\mathcal{H}t}$ .

**Remarks:**

- If the  $\mathcal{H}_i$ 's commute, then we do have  $e^{-i\mathcal{H}t} = \prod_k e^{-i\mathcal{H}_k t}$ .
- The restriction of  $\mathcal{H}_k$  to involve at most  $c$  particles *implies* that in the sum of the Hamiltonians,  $L$  is upper bounded by a polynomial in  $n$ .

The heart of quantum simulation algorithms is the following asymptotic approximation theorem:

**Theorem 5.3.1** (Trotter formula). Let  $A$  and  $B$  be Hermitian operators. Then for any real  $t$ ,

$$\lim_{n \rightarrow \infty} \left( e^{-iAt/n} e^{iBt/n} \right) = e^{i(A+B)t}. \quad (5.28)$$

□

Of course, the theorem holds even if  $A$  and  $B$  don't commute. For now, we only consider the cases where  $A, B$  are Hermitian. But  $A, B$  can actually be generators of certain kinds of semigroups, which correspond to general quantum operations.

Using the theorem (and the proof of the theorem, which I won't show here), we can get

$$e^{i(A+B)\Delta t} = e^{iA\Delta t} e^{iB\Delta t} + \mathcal{O}(\Delta t^2). \quad (5.29)$$

Similarly,

$$e^{i(A+B)\Delta t} = e^{iA\Delta t/2} e^{iB\Delta t} e^{iA\Delta t/2} + \mathcal{O}(\Delta t^3). \quad (5.30)$$

### 5.3.3 The algorithm

- **Inputs:** (1) A Hamiltonian  $\mathcal{H} = \sum_k \mathcal{H}_k$  acting on an  $N$ -dimensional system, where each  $\mathcal{H}_k$  acts on a small subsystem of size independent of  $N$ , (2) an initial state  $|\psi_0\rangle$ , of the system at  $t = 0$ , (3) a positive, non-zero accuracy  $\delta$ , and (3) a time  $t_f$  at which the evolved state is desired.

- A state  $|\tilde{\psi}(t_f)\rangle$  such that

$$\left| \langle \tilde{\psi}(t_f) | e^{-i\mathcal{H}t_f} | \psi_0 \rangle \right|^2 \geq 1 - \delta. \quad (5.31)$$

- Choose a representation such that the state  $|\tilde{\psi}\rangle$  of  $n = \text{poly}(\log N)$  qubits approximates the system and the operators  $e^{-\mathcal{H}_k \Delta t}$  have efficient quantum circuit approximations. Selection an approximation method and  $\Delta t$  such that the expected error is acceptable, construct the corresponding quantum circuit  $\mathcal{U}_{\Delta t}$  for the iterative step, and do the following:

- Initialize state:  $|\tilde{\psi}_0\rangle \leftarrow |\psi_0\rangle; j = 0$ .
- Iterative update:  $\rightarrow |\tilde{\psi}_{j+1}\rangle = \mathcal{U}_{\Delta t} |\tilde{\psi}_j\rangle$ .
- Loop:  $\rightarrow j = j + 1$ ; go to the second step until  $j\Delta t \geq t_f$ .
- Final result:  $\rightarrow |\tilde{\psi}(t_f)\rangle = |\tilde{\psi}\rangle$ .

**Theorem 5.3.2** (Baker-Campbell-Hausdorf formula).

$$e^{(A+B)\Delta t} = e^{A\Delta t} e^{B\Delta t} e^{-(1/2)[A,B]\Delta t^2} + \mathcal{O}(\Delta t^3). \quad (5.32)$$

### 5.3.4 An illustrative example

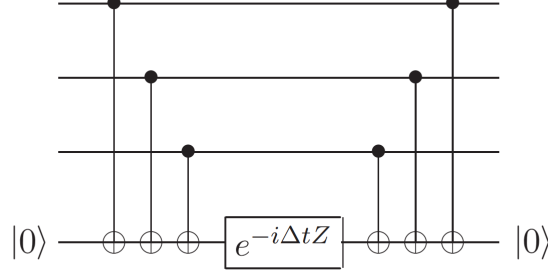
We have seen the case where the Hamiltonian is a sum of local interactions. However, this is not a fundamental requirement. Efficient quantum simulations are possible even for Hamiltonians which act non-trivially on all or nearly all parts of a large system.

Consider the Hamiltonian

$$\mathcal{H} = Z_1 \otimes Z_2 \otimes \cdots \otimes Z_n \quad (5.33)$$

which acts on an  $n$  qubit system. This can be simulated efficiently. What we desire is a simple quantum circuit which implements  $e^{-i\mathcal{H}\Delta t}$ , for arbitrary values of  $\Delta t$ . A circuit that does this is given by

Even though the Hamiltonian involves all the qubits in the system, it does so in a *classical* manner: the phase shift applied to the system is  $e^{-i\Delta t}$  if the *parity* of the  $n$  qubits in the computational basis is even. else the phase shift would

Figure 4.19. Quantum circuit for simulating the Hamiltonian  $H = Z_1 \otimes Z_2 \otimes Z_3$  for time  $\Delta t$ .

be  $e^{i\Delta t}$ . Thus, a simple simulation of  $\mathcal{H}$  is possible by first classically computing the parity, then applying the appropriate phase shift conditioned on the parity, then uncomputing the parity.

Extending the same procedure allows us to simulate more complicated extended Hamiltonians. Specifically, we can efficiently simulate any Hamiltonian of the form

$$\mathcal{H} = \bigotimes_{k=1}^n \sigma_{c(k)}^k \quad (5.34)$$

where  $\sigma_{c(k)}^k$  is a Pauli matrix (or identity) acting on the  $k$ th qubit, with  $c(k) \in \{0, 1, 2, 3\}$  specifying one of  $\{I, X, Y, Z\}$ . The qubits upon which the identity acts can be disregarded, and  $X$  or  $Y$  terms can be transformed by single qubit gates to  $Z$  operations. This leaves a Hamiltonian of the form given above, which can be efficiently simulated.

Using this procedure allows us to simulate a wide class of Hamiltonians containing terms which are not local. It is possible to simulate a Hamiltonian of the form  $\mathcal{H} = \sum_{k=1}^L \mathcal{H}_k$  where we only require that the individual  $\mathcal{H}_k$  have a tensor product structure, and that  $L$  is polynomial in the total number of particles  $n$ .

### 5.3.5 Perspectives on quantum simulation

Quantum simulation is similar to classical methods, but differs fundamentally in many aspects. Each iteration of the quantum algorithm must completely replace the old state with a new one and there is no way to obtain any nontrivial information from an intermediate step without changing the algorithm (due to the no-cloning theorem). Furthermore, the final measurement must be chosen to provide the desired result because it collapses the entire state.

Another difficult problem is the simulation of equilibration processes. A system with Hamiltonian  $\mathcal{H}$  in contact with an environment at temperature  $T$

will come to thermal equilibrium in a state known as the *Gibbs* state,

$$\rho_{\text{therm}} = \frac{e^{-\mathcal{H}/k_B T}}{\mathcal{Z}} \equiv \frac{e^{-\mathcal{H}/k_B T}}{\text{tr}(e^{-\mathcal{H}/k_B T})} \quad (5.35)$$

where of course  $\mathcal{Z}$  is the partition function. We can check that  $\text{tr}(\rho) = 1$ . For now, we don't know how a quantum computer can simulate this.

Indistinguishability of particles also places a constraint on the state vector of a system which manifests itself in two ways: bosons and fermions. The state vector of a system of bosons remains unchanged under permutation of any two constituents. System of fermions experience a sign change in their state vector under interchange of any two constituents. Both kinds of systems can be simulate efficiently on a quantum computer.



## 5.4 Problems in Quantum Simulations and Quantum Algorithms

In this section, we will be looking at the following [review](#) on quantum algorithms for quantum chemistry and materials science.

### 5.4.1 Quantum Simulation: An Overview

#### Introduction

The interest in quantum computing for quantum simulations of molecules and materials stems from the fact that in many cases, the chemistry and physics of molecules and materials is best described using quantum mechanics. In the worst case, quantum simulation is exponentially hard on classical computers. Quantum computers have advantages over classical computers, but these advantages are problem-specific.

As we have seen before, the natural problem to solve on a quantum computer is the time evolution of a quantum system given some initial state:

$$i\partial_t |\psi(t)\rangle = \mathcal{H} |\psi(t)\rangle. \quad (5.36)$$

This problem, as we have pointed out, is of polynomial cost on a quantum computer, which can offer exponential speedup over classical computers. However, it is necessary to prepare the initial state, which may be difficult. In particular, preparing a low-energy state may be challenging, which naturally leads to considering other important problems:

$$\text{Ground state: } \mathcal{H} |\psi_0\rangle = E_0 |\psi_0\rangle; \quad E_0 = \min_{|\psi\rangle} \langle\psi| \mathcal{H} |\psi\rangle \quad (5.37)$$

$$\text{Thermal averages: } \langle\mathcal{A}\rangle = \frac{\text{tr}(\mathcal{A}e^{-\beta\mathcal{H}})}{\text{tr}(e^{-\beta\mathcal{H}})} \quad (5.38)$$

Ground state determination lies in complexity class QMA, a class of problems *not known to be efficiently solvable in general on a quantum computer*. This means that thermal averages cannot in general be computed efficiently on a quantum computer, since in the limit of zero temperature, this problem reduces to ground-state determination.

To understand quantum advantage in chemistry, condensed matter physics, and quantum materials science, we must be guided by actual empirical data in the form of numerical and theoretical experiments with quantum algorithms and quantum devices on simulation problems of interest. This requires progress, of course.

#### Current quantum architectures

There's *analog quantum computation* and there's *digital quantum computation*. Analog quantum computation was proposed by Feynman. The idea of analog

quantum computation is to build a lattice of spins with tunable interactions and let it simulate physical systems. The limitation, however, is that this is quite limited because it can only simulate a certain set of physical systems. Furthermore, its accuracy is limited.

Digital quantum computation is more general. Digital quantum computation involves qubits and gates. One can show that with a finite set of gates, one can in principle generate arbitrary quantum evolutions to arbitrary precision. Every problem in the complexity class BQP can be mapped into a quantum program.

The third model of quantum computation is *adiabatic quantum computation*. This model is considered equivalent to the digital version, but is less practical. We won't worry about this.

### Building a circuit-based digital quantum computer

The natural enemy of quantum computation is decoherence – the tendency of quantum systems to decay into their classical states. We are current in the *noisy intermediate-scale quantum* (NISQ) era, where qubit technologies (superconducting qubits, ion traps, etc) have reached the point where small devices of a few dozen qubits can be sufficiently isolated from decoherence to execute non-trivial quantum algorithms of a few tens to hundreds of gates. With these devices, an artificial but well-defined problem that is intractable classically is solved on a quantum computer. However, these problems are often not practical. The more important question is when a more relevant problem can be solved on a quantum computer.

To address large-scale problems, *quantum error-correction* (QEC) is necessary. QEC leads to a distinction between *logical* and *physical* qubits. The former are error-corrected and encoded in the state of many physical qubits. Quantum algorithms are performed on the logical qubits, and the error-correction scheme translates the operations on logical qubits into physical operations.

### 5.4.2 Simulation challenges in molecular and materials science

In this section we look at some scientific problems of relevance to quantum simulation.

#### Quantum chemistry

Quantum chemistry is concerned with determining the low-lying eigenstates of the electronic Hamiltonian of a molecule. The eigenstates are determined for fixed sets of nuclear positions within the Born-Oppenheimer approximation. Determining the electronic energy as a function of nuclear position is key to understanding chemical reactivity, product distribution, and reaction rates.

Some examples of problems in quantum chemistry include

- The chemistry of enzyme active sites
- Transition metal nanocatalysts and surface catalysts
- Light harvesting and the vision process

The basic metric is whether ground-states or low-energy eigenstate quantum algorithms yield more accurate energies than the best classical algorithms, for the problem sizes of interest.

#### Quantum molecular spectroscopy

The theoretical goal is to compute the eigenstates of the nuclear SE. There are challenges, of course. First, the nuclear Hamiltonian is often not known, because interactions are mediated by electrons. Second, nuclear rotational and vibrational motion is often far from harmonic and not well approximated by simple mean-field theories. Third, even when the nuclear SE has been properly formulated, one faces the challenge of representing the eigenstates.

Some famous examples include:

- Spectra of floppy molecules
- Hydrogen bonded clusters

There are similarities with the quantum chemistry problems, but the differences are more significant. The Hamiltonian is often more complicated because interactions are now many-body. Also, the number of states is often orders-of-magnitude greater.

**Chemical quantum dynamics**

Chemical quantum dynamics is concerned with modeling time-dependent electronic and nuclear quantum effects in molecules. Some problems include:

- Proton coupled electron transfer
- Vibrational dynamics in complex environments
- Plasmonic chemistry

There are multiple practical challenges. Some of this may be viewed as an issue of representation. In addition, the dynamical quantum state involves near continuum degrees of freedom, which makes discretizing the Hilbert space a challenging problem.

**Correlated electronic structure in materials**

The goal here is to determine low-energy properties of materials such as Mott insulators. Some famous problems include:

- High-temperature superconductivity
- Non-Fermi liquid behavior in heavy fermion compounds and fermionic systems near criticality
- Quantum fluctuations in two-dimensional systems
- Frustrated spin systems

### 5.4.3 Challenges for quantum algorithms in quantum simulation

#### Overview of algorithms

In this section, we look at the current status and theoretical and practical challenges to implement quantum algorithms for quantum problems.

The first step in a quantum simulation is to choose a *representation* for the Hamiltonian and the states. We will examine the possibilities for different qubit representations, and open questions.

Quantum ground-state algorithms fall into different classes. QPE is a direct route to nearly exact eigenstate determination, but has been challenging to implement in the near-term era. On the other hand, *variational quantum algorithms* provide the possibility to introduce approximations with adjustable circuit depth, which are determined via optimization, usually implemented in a hybrid-quantum classical loop.

While polynomial time algorithms for *quantum time evolution* have been known for some time, algorithms with optimal asymptotic complexity as well as favorable scaling with error and low prefactors, remain an area of active research.

#### Qubit representation of many-body systems

To simulate many-body systems on a digital computer (either quantum or classical), the infinite-dimensional Hilbert space of a many-body system has to be truncated. The most direct route then is to define a finite set of *basis functions* and then try to project the exact many-body Hamiltonian onto the chosen basis. The resulting discretized system is then expressed in terms of qubits. A choice of a good representation is important as it may affect the simulation cost dramatically.

##### 1. *Ab initio electronic structure qubit representations:*

This concerns with the electronic structure. Consider the Hamiltonian that appears regularly in chemistry and physics

$$\mathcal{H} = \sum_{i=1}^K -\frac{1}{2}\nabla^2 + V(r_i) + \sum_{1 \leq i < j \leq K} \frac{1}{|r_i - r_j|} \quad (5.39)$$

where  $K$  is the number of electrons,  $V$  is the potential created by atomic nuclei at a point  $r$ , and the last term represents the Coulomb interaction. Each electronic has  $r_i \in \mathbb{R}^3$  and spin  $\omega_i \in \{\uparrow, \downarrow\}$ . So, the total wavefunction (for  $K$  electrons) has the form  $\psi = \psi(\vec{x}_1, \dots, \vec{x}_K)$  where  $\vec{x}_i = (r_i, \omega_i)$ . From Fermi statistic, we require that  $\psi$  is anti-symmetric under a position

swap.

The first step is to approximate the electronic Hamiltonian with a simpler simulator Hamiltonian. This is usually achieved by truncate the Hilbert space of a single electron to a finite set of basis functions  $\psi_1, \dots, \psi_N$ .

Electronic structure simulation algorithms are based on the *first quantization method* ([link](#)). We won't worry about this too much.

The *second quantization method* often results in simpler simulator Hamiltonian and requires few qubits. This methods is particularly well suited for quantum simulation algorithms and has been experimentally demonstrated for small molecules. Given a set of  $N$  basis functions  $\psi_1, \dots, \psi_N$ , the second quantized simulator Hamiltonian is given by

$$\mathcal{H} = \sum_{p,q=1}^N t_{pq} \hat{c}_p^\dagger \hat{c}_q + \frac{1}{2} \sum_{p,q,r,s=1}^N u_{pqrs} \hat{c}_p^\dagger \hat{c}_q^\dagger \hat{c}_r \hat{c}_s. \quad (5.40)$$

Again, we're not going to worry too much about these.

## 2. Electronic basis functions:

The discretization of the electronic Hilbert space for a quantum simulation requires balancing two concerns. We need to represent the state with the smallest number of qubits, but also retain maximal Hamiltonian sparsity.

There are two families of basis functions in wide use in quantum chemistry and quantum materials science: atomic orbital *Gaussian bases* and *Plane waves*. Gaussian bases are commonly used in molecular simulations due to their compactness, while plane waves are used in crystalline materials simulations.

In Gaussian bases, linear combinations of Gaussian functions are placed at the nuclear positions. When they are place where the ground-state electron density is highest, they give a compact representation of the wavefunction for bound states. However, the Hamiltonian is not sparse, which leads to high gate counts.

Plane waves offer greater simplicity as the accuracy of the basis is controlled by a single parameter, namely the kinetic energy cutoff. The number of plane waves needed to reach a desired accuracy is larger than th number of Gaussian states, but the Hamiltonian contains fewer terms ue to momentum conservation.

The need to expose more sparsity in the Hamiltonian while retaining a reasonable compact wavefunction is an active area of research in both classical and quantum algorithms.

### 3. *Fermions-to-qubit mappings:*

Since the basic units of a quantum computer are qubits rather than fermions, any quantum simulation algorithm of fermions employs a suitable encoding of fermionic degrees of freedom into qubits. The standard *Jordan-Wigner* mapping identifies each Fermi mode with a qubit such that the empty and the occupied states are mapped to the qubit basis states  $|0\rangle$  and  $|1\rangle$  respectively. The Jordan-Wigner mapping/transformation is a transformation that maps spin operators onto fermionic creation and annihilation operators. A natural question is whether the number of

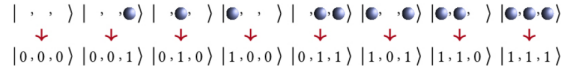


FIG. 10. Schematic of the Jordan-Wigner encoding for 3 spin-orbitals. Fermions are represented by blue spheres. Each Fock basis vectors (from the vacuum state, left, to the completely filled state, right) is mapped onto a 3-qubit state, with empty (filled) spin-orbitals corresponding to qubits in 0 (1).

qubits required to express a Fermi system can be reduced by exploiting symmetries such as the particle number conservation or the point group symmetries of molecules. For example, zero temperature simulations often target only one symmetry sector containing the ground state. This motivates the study of *symmetry-adapted fermion-to-qubit mappings*.

## Quantum algorithms for ground and excited states

There are many approaches to obtaining ground states or excited states on a quantum computer. State preparation procedures attempt to construct a circuit to prepare a state with as large as possible overlap with the desired eigenstate. One set of such procedures, which includes adiabatic state preparation and quantum imaginary time evolution, uses a prescribe evolution path. An alternative strategy is based on *variational methods*, which are often called variational quantum eigensolvers. Here, the preparation circuit itself is defined via the optimization of the energy with respect to parameters of the circuit.

Given some state with sufficiently large overlap with the desired state, one can perform *quantum phase estimation* (QPE) which simultaneously projects the state onto an eigenstate of the Hamiltonian and obtains an estimate for the energy of this eigenstate. The error, as we have seen, is inversely proportional to the simulation time. The probability of successfully projecting onto the desired state is given by the square overlap of the input state and the desired state, and

it is thus necessary to use some other method (such as the state preparation procedures above) to prepare an input state with sufficient overlap with the desired state.

There are weaknesses to this, of course. While phase estimation allows the deviation of the final state from an exact eigenstate to be systematically reduced, it can require deep circuits with many controlled gates that are challenging for devices with limited coherence and without error correction. Variational methods replace such circuits by a large number of potentially short simulations. However, if one does not measure the energy by phase estimation, but instead by expressing the Hamiltonian as a sum of multi-qubit Pauli operators and measuring the terms individually, the state preparation and measurements must be repeated many times, with the error converging only as the square root of the number of repetitions. Variational methods are also limited by the variational form and ability to solve the associated optimization problem, which may by itself represent a difficult classical optimization.

### Preparing ground states along a prescribed path

#### 1. *Adiabatic state preparation:*

This state preparation relies on the *adiabatic theorem*

**Theorem 5.4.1** (Adiabatic Theorem). A physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum.

In simpler terms, this says that gradually changing conditions allow the system to adapt its configuration, hence the probability density is modified by the process. If the system started in an eigenstate of the initial Hamiltonian, it will end in the *corresponding* eigenstate of the final Hamiltonian.

To make use of this theorem, one chooses a Hamiltonian path  $\mathcal{H}(\lambda)$  where  $\lambda \in [0, 1]$ , such that the ground state of  $\mathcal{H}(0)$  is easily prepared, while  $\mathcal{H}(1)$  is the Hamiltonian whose ground state one wants to obtain. The system is then evolved under the time-dependent SE

$$i\partial_t |\psi_t\rangle = \mathcal{H}(t/T) |\psi_t\rangle, \quad t \in [0, T]. \quad (5.41)$$

When  $T \rightarrow \infty$  and if the spectrum of  $\mathcal{H}_{t/T}$  is gapped for all  $t$ , the final state is the exact ground state of  $\mathcal{H}$ . Away from the adiabatic limit  $T \rightarrow \infty$ , corrections are necessary.

The main disadvantage of this approach is that it is limited by the  $T \rightarrow \infty$  requirement, and the fact that a path without degeneracies must be chosen.



2. *Quantum imaginary-time evolution:*

In classical simulations, one popular approach to prepare (nearly exact) ground-states is imaginary-time evolution, which expresses the ground-state as the long-time limit of the imaginary-time SE  $-\partial_t |\Phi(\beta)\rangle = \mathcal{H} |\Phi(\beta)\rangle$ :

$$|\psi\rangle = \lim_{\beta \rightarrow \infty} \frac{e^{-\beta \mathcal{H}} |\Phi(\beta)\rangle}{\|e^{-\beta \mathcal{H}} \Phi(\beta)\|}. \quad (5.42)$$

To perform imaginary time evolution on a quantum computer, it is necessary to implement the (repeated) action of the short-imaginary-time propagator  $e^{-\Delta\tau \mathcal{H}}$  on a state. Given a Hamiltonian that can be decomposed into geometrically local terms,

$$\mathcal{H} = \sum_m \hat{h}_m \quad (5.43)$$

and a state  $|\psi\rangle$  with finite correlation length  $C$ , the action  $e^{-\Delta\tau \hat{h}_i}$  can be generated by a unitary  $\mathcal{U} = e^{i\hat{A}}$  acting on  $\mathcal{O}(C)$  qubits surrounding those acted on by  $\hat{h}_i$ , i.e.,

$$\frac{e^{-\beta \mathcal{H}} |\psi\rangle}{\|e^{-\beta \mathcal{H}} \psi\|} = \mathcal{U} |\psi\rangle = e^{i\hat{A}} |\psi\rangle \quad (5.44)$$

where the coefficients of the Pauli strings in  $\hat{A}$  can be determined from local measurements of the qubits around  $\hat{h}_i$ .

Quantum imaginary time evolution becomes inefficient in terms of the number of measurements and complexity of the operator  $\hat{A}$  if the domain  $C$  grows to be large along the imaginary time evolution path.

**Variational state preparation and variational quantum eigensolver**

States can also be prepared through variational methods. Here, similar to classical variational approaches, one chooses a class of ansatz states for the ground state of the Hamiltonian of interest. Generally speaking, such an ansatz consists of some initial state and a unitary circuit parametrized by some set of classical variational parameters. Applying this circuit to the initial state yields a guess for the ground state, whose energy is then evaluated. This yields an upper bound to the true ground state energy. One then varies the variational parameters to lower the energy of the ansatz state.

Choosing the class of ansatz states is also a balancing act. On the one hand, the class should contain an accurate approximation to the true ground state of the system. On the other hand, we want a class of circuits that are easily executed on the available quantum computer. Finally, it is important for the classical optimization over the variational parameters to be well-behaved, so as to be able to find low-energy minima.

## 1. Unitary coupled cluster:

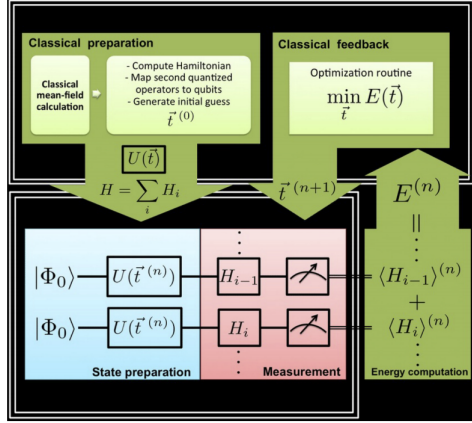


FIG. 13. Workflow of the variational quantum eigensolver algorithm. The classical optimization routine adds expectation values of the Hamiltonian Pauli terms to calculate the energy and estimates new values for the unitary parameters. The process is repeated until convergence. From [172].

We won't worry about the details of how this works at this point. For more information, consider this [paper](#).

## 2. Adapt-VQE ansatz:

In the adapt-VQE scheme, a collection of operators  $\hat{A}_i$  is chosen in advance, and the ground state is approximated by

$$|\psi_{\text{adapt-VQE}}\rangle = e^{\theta_n \hat{A}_n} \dots e^{\theta_1 \hat{A}_1} |\psi_{\text{HF}}\rangle, \quad (5.45)$$

Given a current parameter configuration  $\vec{\theta}$ , the commutator of the Hamiltonian with each operator in the pool is measured to obtain the gradient of the energy

$$E = \langle \psi_{\text{adapt-VQE}} | \mathcal{H} | \psi_{\text{adapt-VQE}} \rangle. \quad (5.46)$$

wrt the parameters  $\vec{\theta}$ . Repeating this multiple times and averaging over the obtained samples gives the gradient of the expectation value of the Hamiltonian wrt to the coefficient of each operator. The ansatz is improved by adding the operator  $\hat{A}_i$  with the largest gradient to the left end of the ansatz with a new variational parameter, thereby increasing  $n$ . The operation is repeated until convergence.

### 3. Other considerations:

There are other considerations with regards to variational methods, but we won't worry about them too much for now.

### Excited states

While much of the above discussion of state preparation and variational algorithms has focused on ground-states, most of the same methods can also be used with minor extensions for excited states. For example, adiabatic state preparation can be used to prepare an excited state, so long as it is connected to the initial state without a vanishing gap.

### Phase estimation

We have a section on this, so let's just be brief and gloss over the details.

Quantum Phase Estimation (QPE) is a crucial step in many quantum algorithms. QPE enables high-precision measurements of the ground and excited energy levels. This is done by preparing a trial initial state  $|\psi(0)\rangle$  that has a non-negligible overlap with the relevant eigenvector of the target Hamiltonian  $\mathcal{H}$  and applying a circuit that creates a superposition of time evolved states  $|\psi(t)\rangle = e^{-i\mathcal{H}t} |\psi(0)\rangle$  over a suitable range of evolution times  $t$ .

Since QPE is used ubiquitously in a variety of quantum applications, it is crucial to optimize its performance. Below we list some open problems that are being actively investigated:

- Given limitations of near-term quantum devices, of particular interest are tradeoffs between the depth of the QPE circuit and its spectral-resolution power as well as its sensitivity to noise.
- Several methods have been proposed for mitigating experimental errors for VQE-type simulations. Such methods enable reliable estimation of expected values of observables on a given trial state without introducing any overhead in terms of extra qubits or quantum gates.
- Classical post-processing methods that enable simultaneous estimation of multiple eigenvalues are highly desirable.
- Finally, a natural question is whether the time evolution operator  $e^{-i\mathcal{H}t}$  in QPE can be replaced by some other functions of  $\mathcal{H}$  that are easier to implement

### Quantum algorithms for time evolution

1. *Hamiltonian simulation problem:*

A quantum computer can be programmed to efficiently simulate the unitary time evolution of almost any physically realistic quantum system. Consider any evolution which is of course given by the SE

$$i\partial_t |\psi(t)\rangle = \mathcal{H} |\psi(t)\rangle, \quad t \geq 0. \quad (5.47)$$

We can just assume that  $\mathcal{H}$  describes a system of  $n$  qubits, since any fermionic or spin system can be mapped to qubits. Integrating we get

$$|\psi(t)\rangle = e^{-it\mathcal{H}} |\psi(0)\rangle. \quad (5.48)$$

A quantum algorithm for Hamiltonian simulation takes as input a description of  $\mathcal{H}$ , the evolution time  $t$ , and outputs a quantum circuit  $\mathcal{U}$  that approximates the time evolution operator  $e^{-it\mathcal{H}}$  within a specified precision  $\epsilon$ :

$$\|\mathcal{U} - e^{-it\mathcal{H}}\| \leq \epsilon. \quad (5.49)$$

$\mathcal{U}$  may use ancillary qubits initialized in the  $|0\rangle$  state. The simulation cost is usually quantified by the runtime of the algorithm (gate count) and the total number of qubits. Applying  $\mathcal{U}$  to  $|\psi(0)\rangle$  gives the approximation state  $|\psi(t)\rangle$ .

Quantum simulation algorithms apply to general classes of Hamiltonians satisfying mild technical conditions that enable a quantum algorithm to access the Hamiltonian efficiently.

## **5.5 Some Models in Quantum Statistical & Condensed-Matter Physics**

### **5.5.1 The Hubbard Model**

### **5.5.2 The Ising Model**

## 5.6 Quantum Phase Estimation (QPE)

One of the most important applications of the Quantum Fourier Transform is the Quantum Phase Estimation. Phase estimation is the key for many quantum algorithms. Suppose a unitary operator  $\mathcal{U}$  has an eigenvector  $|u\rangle$  with eigenvalue  $2^{2\pi i\varphi}$ , where the value of  $\varphi$  is unknown. The goal of the phase estimation algorithm is to estimate  $0 \leq \varphi \leq 1$  with high probability within additive error  $\epsilon$ . The algorithm uses  $\mathcal{O}(\log(1/\epsilon))$  qubits and  $\mathcal{O}(1/\epsilon)$  controlled- $\mathcal{U}$  operations.

To perform the estimation we assume that we have available *oracles* capable of preparing the state  $|u\rangle$  (the eigenstate) and performing the controlled- $\mathcal{U}^{2^j}$  operation, for suitable nonnegative integers  $j$ . Since preparing eigenstates is not necessarily easy, the phase estimation algorithm is not a complete algorithm by itself. Rather, we should view it as a “subroutine” or a “module.”

The QPE uses two registers. The first register contains  $t$  qubits initially in  $|0\rangle$ . The choice of  $t$  depends on two things: the number of digits of accuracy we wish to have in our estimate of  $\varphi$ , and with what probability we wish the phase estimation procedure to be successful. The second register begins in the eigenstate  $|u\rangle$ , and contains as many qubits as necessary to store  $|u\rangle$ .

QPE occurs in two main stages. First, the circuit applies an  $\mathcal{H}$  to each of the  $t$  qubits. Then, the circuit applies controlled- $\mathcal{U}$  operators on the second register, with  $\mathcal{U}$  raised to successive powers of two. This stage looks like the following: Recall that immediately after  $\mathcal{H}^{\otimes t}$  the state of the upper register looks like

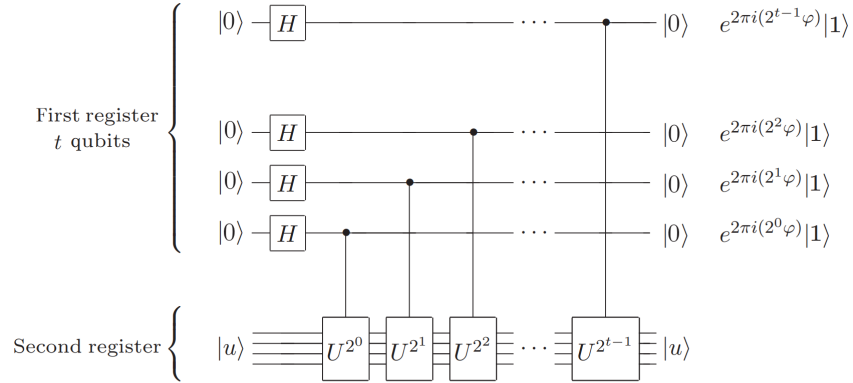


Figure 5.2. The first stage of the phase estimation procedure. Normalization factors of  $1/\sqrt{2}$  have been omitted, on the right.

Figure 5.1: Source: Mike and Ike

$$\mathcal{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad (5.50)$$

The controlled- $\mathcal{U}$  acts on  $|u\rangle$  only if the control qubit is in  $|1\rangle$ . With this, we see that the final state of the first register is given by

$$\begin{aligned} & \frac{1}{2^{t/2}} \left( |0\rangle + e^{2\pi i 2^{t-1} \varphi} |1\rangle \right) \otimes \left( |0\rangle + e^{2\pi i 2^{t-2} \varphi} |1\rangle \right) \otimes \cdots \otimes \left( |0\rangle + e^{2\pi i 2^0 \varphi} |1\rangle \right) \\ &= \dots \\ &= \frac{1}{2^{t/2}} \sum_{0 \leq k < 2^t} e^{2\pi i \varphi k} |k\rangle_t. \end{aligned} \quad (5.51)$$

**Example 5.6.1.** For  $t = 2$ , we have

$$\begin{aligned} & \frac{1}{2} \left( |0\rangle + e^{4\pi i \varphi} |1\rangle \right) \otimes \left( |0\rangle + e^{2\pi i \varphi} |1\rangle \right) \\ &= \frac{1}{2} \left( |0\rangle_2 + e^{1 \cdot 2\pi i \varphi} |1\rangle_2 + e^{2 \cdot 2\pi i \varphi} |2\rangle_2 + e^{3 \cdot 2\pi i \varphi} |3\rangle_2 \right) \\ &= \frac{1}{2^{2/2}} \sum_{0 \leq k < 2^2} e^{2\pi i \varphi \cdot 2} |k\rangle_2 \end{aligned} \quad (5.52)$$

□

Note that we're leaving out the qubits carrying the  $|u\rangle$  state because we don't do anything to it (because  $|u\rangle$  is an eigenstate of  $\mathcal{U}$ ). We should also note that the effect of the sequence of controlled- $\mathcal{U}$  operations is to take  $|j\rangle |u\rangle$  to  $|j\rangle \mathcal{U}^j |u\rangle$ . We should also note that this does not depend on the fact that  $|u\rangle$  is an eigenstate of  $\mathcal{U}$ .

In the second state, the algorithm applies the inverse quantum Fourier transform on the first register (which is just the QFT, applied in reverse order of operations). This can be done efficiently in  $\mathcal{O}(t^2)$  steps, as we have seen the Quantum Information [notes](#). Since the input in this case is the maximal superposition in the  $t$ -vector space, the output of the inverse QFT is going to be a single state:  $|\tilde{\varphi}\rangle$ , which we hope is approximately  $|\varphi\rangle$ . We will see how close  $|\tilde{\varphi}\rangle$  can get to  $|\varphi\rangle$ .

The third and final stage of the algorithm is to read out the state of the first register by doing a measurement in the computational basis.

**Example 5.6.2.** To make sure we see what is going on, suppose  $\varphi$  is expressed exactly in  $t$  qubits, as  $\varphi = 0, \varphi_1 \varphi - 2 \dots \varphi_t$  (note that  $\varphi \in [0, 1]$ ). Then the state of the input register after the first stage is given by

$$\frac{1}{2^{t/2}} \left( |0\rangle + e^{2\pi i 0 \cdot \varphi_t} |1\rangle \right) \otimes \left( |0\rangle + e^{2\pi i 0 \cdot \varphi_{t-1} \varphi_t} |1\rangle \right) \otimes \cdots \otimes \left( |0\rangle + e^{2\pi i 0 \cdot \varphi_1 \varphi_2 \dots \varphi_t} |1\rangle \right). \quad (5.53)$$

Applying the inverse QFT to this, we see that output state from the second state is the product state  $|\varphi_1 \dots \varphi_t\rangle$ . A measurement in the computational basis gives  $\varphi$  exactly. □

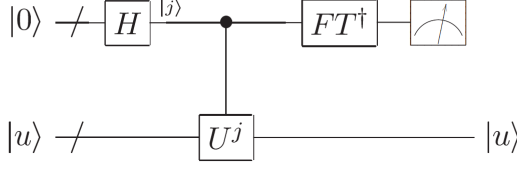


Figure 5.3. Schematic of the overall phase estimation procedure. The top  $t$  qubits (the ‘/’ denotes a bundle of wires, as usual) are the first register, and the bottom qubits are the second register, numbering as many as required to perform  $U$ .  $|u\rangle$  is an eigenstate of  $U$  with eigenvalue  $e^{2\pi i\varphi}$ . The output of the measurement is an approximation to  $\varphi$  accurate to  $t - \lceil \log(2 + \frac{1}{2\epsilon}) \rceil$  bits, with probability of success at least  $1 - \epsilon$ .

Figure 5.2: Source: Mike and Ike

Schematically, the algorithm for QPE looks like the following:

In summary, what we basically did was using the inverse QFT to get

$$\frac{1}{2^{t/2}} \sum_{0 \leq k < 2^t} e^{2\pi i \varphi j} |j\rangle |u\rangle \rightarrow |\tilde{\varphi}\rangle |u\rangle \quad (5.54)$$

where  $|\tilde{\varphi}\rangle$  is a good estimator for  $\varphi$  when measured.

### 5.6.1 Performance and Requirements

What happens when  $\varphi$  cannot be written exactly with a  $t$ -bit expansion? We will now show that the procedure we described will produce a good approximation to  $\varphi$  with high probability.

Let  $b \in [0, 2^t - 1]$  an integer be given such that  $b/2^t = 0.b_1b_2 \cdots b_t$  is the best  $t$  bit approximation to  $\varphi$  which is less than  $\varphi$ , i.e.,

$$0 \leq \delta \equiv \varphi - b/2^t \leq 2^{-t}. \quad (5.55)$$

(This is getting pretty close to an analysis proof – because it *is* an analysis proof.) Applying the inverse QFT to the state immediately after the first stage gives

$$\begin{aligned} \mathcal{F}^{-1} \left[ \frac{1}{2^{t/2}} \sum_{0 \leq k < 2^t} e^{2\pi i \varphi k} |k\rangle \right] &= \frac{1}{2^t} \sum_{0 \leq k, l < 2^t} e^{\frac{-2\pi i k l}{2^t}} e^{2\pi i \varphi k} |l\rangle \\ &= \frac{1}{2^t} \sum_{0 \leq k, l < 2^t} \left( e^{2\pi i (\varphi - l/2^t)} \right)^k |l\rangle \end{aligned} \quad (5.56)$$

by definition. Note that when  $\varphi$  can be expressed using exactly  $t$  bits we are reduced to the simpler version of the problem described above.



Let  $\alpha_l$  be the amplitude of  $|(b+l) \bmod 2^t\rangle$ , i.e.,

$$\alpha_l \equiv \frac{1}{2^t} \sum_{0 \leq k < 2^t} \left( e^{2\pi i(\varphi - (b+l)/2^t)} \right)^k. \quad (5.57)$$

This is a partial sum of a geometric series. It is straightforward to show that

$$\alpha_l = \frac{1}{2^t} \left( \frac{1 - e^{2\pi i(2^t \varphi - (b+l))}}{1 - e^{2\pi i(\varphi - (b+l)/2^t)}} \right) = \frac{1}{2^t} \left( \frac{1 - e^{2\pi i(2^t \delta - l)}}{1 - e^{2\pi i(\delta - l/2^t)}} \right). \quad (5.58)$$

Suppose the outcome of the final measurement is  $m$ . We want to bound the probability that  $|m - b| > \epsilon$  (where  $\epsilon$  is the tolerance), i.e., we don't want the probability that  $m$  is far from  $b$  to be large. Well, the probability of observing such an  $m$  is given by

$$P(|m - b| > \epsilon) = \sum_{-2^{t-1} < l \leq -(\epsilon+1)} |\alpha_l|^2 + \sum_{\epsilon+1 \leq l \leq 2^{t-1}} |\alpha_l|^2. \quad (5.59)$$

For any real  $\theta$ , we have  $|1 - \exp(i\theta)| \leq 2$ , so

$$|\alpha_l| \leq \frac{2}{2^t |1 - e^{2\pi i(\delta - l/2^t)}|}. \quad (5.60)$$

Further, provided  $-\pi \leq \theta \leq \pi$ , we have  $|1 - \exp(i\theta)| \geq 2|\theta|/\pi$ . Also, when  $-2^{t-1} < l \leq 2^{t-1}$  we have  $-\pi \leq 2\pi(\delta - l/2^t) \leq \pi$ , so we have

$$|\alpha_l| \leq \frac{1}{2^{t+1}(\delta - l/2^t)}. \quad (5.61)$$

With this we have

$$\begin{aligned} p(|m - b| > \epsilon) &\leq \frac{1}{4} \left( \sum_{l=-2^{t-1}+1}^{-(\epsilon+1)} \frac{1}{(l - 2^t \delta)^2} + \sum_{l=\epsilon+1}^{2^{t-1}} \frac{1}{(l - 2^t \delta)^2} \right) \\ &\leq \frac{1}{4} \left( \sum_{l=-2^{t-1}+1}^{-(\epsilon+1)} \frac{1}{(l-1)^2} + \sum_{l=\epsilon+1}^{2^{t-1}} \frac{1}{(l-1)^2} \right), \quad 0 \leq 2^t \delta < 1 \\ &\leq \frac{1}{2} \sum_{l=\epsilon}^{2^{t-1}-1} \frac{1}{l^2} \\ &\leq \frac{1}{2} \int_{\epsilon-1}^{2^{t-1}-1} \frac{1}{l^2} dl \\ &= \frac{1}{2(\epsilon-1)}. \end{aligned} \quad (5.62)$$

Suppose we wish to approximate  $\varphi$  to an accuracy  $2^{-n}$ , i.e., we choose  $\epsilon = 2^{t-n} - 1$ . By using  $t = n + p$  qubits, we see that the probability of obtaining an

approximation correct to this accuracy is at least

$$1 - \frac{1}{2(\epsilon - 1)} = 1 - \frac{1}{2(2^p - 2)}. \quad (5.63)$$

So, to successfully obtain  $\varphi$  accurate to  $n$  bits with probability of success at least  $1 - \epsilon$  we choose

$$t = n + \left\lceil \log \left( 2 + \frac{1}{2\epsilon} \right) \right\rceil. \quad (5.64)$$

Now here's a catch: to use QPE, we need the eigenstate  $|u\rangle$ . What if we don't know how to prepare such a state? Suppose that we prepare some other state  $|\psi\rangle$  in place of  $|u\rangle$ . We can write

$$|\psi\rangle = \sum_u c_u |u\rangle \quad (5.65)$$

where  $|u\rangle$  are the eigenstates of  $U$ . Suppose the eigenstate  $|u\rangle$  has the eigenvalue  $e^{2\pi i \varphi_u}$ . Running the QPE on  $|\psi\rangle$  will give

$$|\psi\rangle \xrightarrow{\text{QPE}} \sum_u c_u |\tilde{\varphi}_u\rangle |u\rangle, \quad (5.66)$$

where  $\tilde{\varphi}_u$  is an estimation for  $\varphi_u$ . Reading out the first register gives us a good approximation for  $\varphi_u$ , where  $u$  is chosen at random with probability  $|c_u|^2$ . This procedure allows us to avoid preparing a possibly unknown eigenstate, but at the cost of introducing some additional randomness into the algorithm. It turns out that if  $t$  is chosen according to the minimization procedure above, then the probability for measuring  $\varphi_u$  accurate to  $n$  bits at the conclusion of the phase estimation algorithm is at least  $|c_u|^2(1 - \epsilon)$ .

### 5.6.2 The algorithm

QPE solves a problem which is both non-trivial and interesting from a physical point of view. It allows us to estimate the eigenvalue associated to a given eigenvector of a unitary operator. It turns out that other interesting problems can be reduced to phase estimation, which makes QPE particularly useful.

Here's the algorithm:

- **Inputs:** (1) An oracle which performs a controlled- $\mathcal{U}^j$  operation, (2) an eigenstate  $|u\rangle$  of  $\mathcal{U}$  with eigenvalue  $e^{2\pi i \varphi_u}$ , and (3)  $t = n + \lceil \log(2 + 1/2\epsilon) \rceil$  qubits initialized to  $|0\rangle$ .
- **Outputs:** An  $n$ -bit approximation  $\tilde{\varphi}_u$  of  $\varphi_u$ .
- **Runtime:**  $\mathcal{O}(t^2)$  operations (due to the inverse QFT) and one call to controlled- $\mathcal{U}^j$  oracle. Succeeds with probability at least  $1 - \epsilon$ .

- **Procedure:**

- Initial state:  $|0\rangle |u\rangle$
- Create superposition:  $\rightarrow (1/2^{t/2}) \sum_{0 \leq j < 2^t} |j\rangle |u\rangle$ .
- Apply the oracle:  
 $\rightarrow (1/2^{t/2}) \sum_{0 \leq j < 2^t} |j\rangle \mathcal{U}^j |u\rangle = (1/2^{t/2}) \sum_{0 \leq j < 2^t} e^{2\pi i j \varphi_u} |j\rangle |u\rangle$
- Apply inverse QFT:  $\rightarrow |\tilde{\varphi}_u\rangle |u\rangle$ .
- Measure the first register:  $\rightarrow \tilde{\varphi}_u$ .

The algorithm has a wide range of applications, including order-finding and factorization.

## 5.7 Quantum Variational Eigensolver (QVE)

In this section we're looking this [paper](#) on the quantum variational eigensolver (QVE).

For quantum systems, where the physical dimension grows exponentially, finding the eigenvalues of certain operators is one of many intractable problems and remains a fundamental challenge. QPE efficiently finds the eigenvalue of a given eigenvector but requires fully coherent evolution. This paper presents an alternative approach that greatly reduces the requirements for coherent evolution and combine this method with a new approach to state preparation based on ansatz and classical optimization.

As we have seen, quantum approaches to finding eigenvalues have previously relied on the quantum phase estimation (QPE) algorithm. The QPE algorithm offers an exponential speedup over classical methods and requires a number of quantum operations  $\mathcal{O}(p^{-1})$  to obtain an estimate with precision  $p$ . Recall that in the standard formulation of QPE, one assumes the eigenvector  $|\psi\rangle$  of a Hermitian operator  $\mathcal{H}$  is given as input and the problem is to determine the corresponding eigenvalue  $\lambda$ . The time the quantum computer must remain coherent is determined by the necessity of  $\mathcal{O}(p^{-1})$  successive applications of  $e^{-i\mathcal{H}t}$ , each of which can require on the order of millions or billions of quantum gates for practical applications, as compared to the tens to hundreds of gates achievable in the short term.

This paper presents an alternative to QPE that significantly reduces the requirements for coherent evolution. This alternative method is given by a quantum processing unit which efficiently calculates the expectation value of a Hamiltonian, providing an exponential speedup over exact diagonalization, the only known exact solution to the problem on a traditional computer. In conjunction with a classical optimization algorithm, this approach reduces the requirement for coherent evolution of the quantum state, making more efficient use of quantum resources, and may offer an alternative route to practical quantum-enhanced computation.

### 5.7.1 Quantum Expectation Estimation (QEE)

QEE computes the expectation value of a given Hamiltonian  $\mathcal{H}$  for an input state  $|\psi\rangle$ . This is done via the decomposition:

$$\mathcal{H} = \sum_{i\alpha} h_{\alpha}^i \sigma_{\alpha}^i + \sum_{ij\alpha\beta} h_{\alpha\beta}^{ij} \sigma_{\alpha}^i \sigma_{\beta}^j + \dots \quad (5.67)$$

for real  $h$ , where Roman indices identify the subsystem on which the operator acts, and Greek indices identify the Pauli operator. For example,  $\alpha = x$ . This is possible because *products of Pauli matrices form a basis for the set of Hermitian*

matrices of dimensions that are powers of 2. In more generality, this is possible because we have the *Pauli group*  $G_n$ , which is the group generated by the operators applied to each of  $n$  qubits in the tensor product Hilbert space  $(\mathbb{C}^2)^{\otimes n}$ .

In any case, I digress. By linearity, we have

$$\langle \mathcal{H} \rangle = \sum_{i\alpha} h_{\alpha}^i \langle \sigma_{\alpha}^i \rangle + \sum_{ij\alpha\beta} h_{\alpha\beta}^{ij} \langle \sigma_{\alpha}^i \sigma_{\beta}^j \rangle + \dots \quad (5.68)$$

Just as in QPE, we only consider Hamiltonians that can be written as a polynomial number of terms, wrt the system size. This class of Hamiltonians encompasses a wide range of physical system, so this requirement is not too strict. In fact, Hamiltonians of this kind include the electronic structure Hamiltonian of quantum chemistry, the quantum Ising Model, the Heisenberg Model, matrices that are well approximate as a sum of  $n$ -fold tensor products, and more generally any  $k$ -sparse Hamiltonian without evident tensor product structure.

**Remark:** Sparse Hamiltonians are important because it is possible to decompose  $\mathcal{H}$  into  $\sum_k \mathcal{H}_k$  where  $\mathcal{H}_k$ 's all commute (making diagonalization straightforward). If the matrix is sparse, then we shouldn't need too many distinct  $\mathcal{H}_k$ 's. Then we can simulate the Hamiltonian evolution

$$e^{-i\mathcal{H}t} = \prod e^{-i\mathcal{H}_m \delta t} \quad (5.69)$$

where  $t = N\delta t$ . Now, what do we mean by “sparse?” Sparse means that there are at most  $d$  nonzero entries per row,  $d = \text{poly}(\log N)$ . In any given row, the location of the  $j$ th nonzero entry and its value can be computed efficiently (or is given by a black box). Sparse Hamiltonians are different from “Local” Hamiltonians, where  $\mathcal{H} = \sum_j \mathcal{H}_j$  where each  $\mathcal{H}_j$  acts on  $\mathcal{O}(1)$  qubits.  $\square$

In any case, I digress again. The evaluation of  $\langle \mathcal{H} \rangle$  reduces to the sum of a polynomial number of expectation values of simple Pauli operators for a quantum state  $|\psi\rangle$ , multiplied by some real constants. A quantum device can efficiently evaluate the expectation value of a tensor product of arbitrary number of simple Pauli operators. Therefore, with an  $n$ -qubit state we can efficiently evaluate the expectation value of this  $2^n \times 2^n$  Hamiltonian.

This is possible with a classical computer, but it suffers from the  $N$ -representability problem. The power of the quantum approach derives from the fact that quantum hardware can store a global quantum state with exponentially fewer resources than required by classical hardware, and as a result the  $N$ -representability problem does not arise.

The expectation value of a tensor product of an arbitrary number of Pauli operators can be estimated by local measurement of each qubit (this can be found somewhere in Mike and Ike). Such independent measurements can be performed in parallel, incurring a constant cost in time. Furthermore, since

these operators are normalized and finite-dimensional, their spectra are bounded. As a result, each

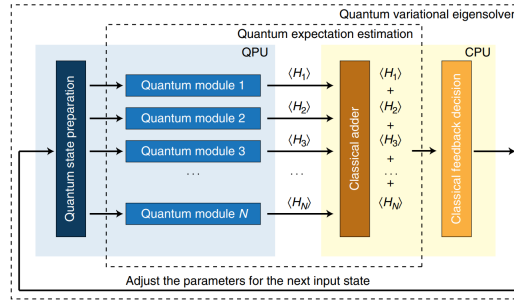
$$\langle H_i^m \rangle = h_{\alpha\beta\dots}^{ij\dots} \langle \sigma_\alpha^i \otimes \sigma_\beta^i \otimes \dots \rangle \quad (5.70)$$

can be estimated to a precision  $p$  of an individual element with coefficient  $h$ , which is an arbitrary element from the set of constants  $\{h_{\alpha\beta\dots}^{ij\dots}\}$  at a cost of  $\mathcal{O}(|h_{\max}|^2 M p^{-2})$  repetitions.  $M$  is the number of terms in the decomposition of the Hamiltonian and  $h_{\max}$  is the coefficient with maximum norm in the decomposition of the Hamiltonian.

The advantage of this approach is that the coherence time to make a single measurement after preparing the state is  $\mathcal{O}(1)$ . Conversely, the disadvantage of this approach with respect to QPE is the scaling in the total number of operations, as a function of the desired precision is quadratically worse  $\mathcal{O}(p^{-2})$  versus  $\mathcal{O}(p^{-1})$ . Moreover, this scaling will also reflect the number of state preparation repetitions required, whereas in QPE the number of state preparation steps is constant. In essence, we dramatically reduce the coherence time requirement while maintaining an exponential advantage over the classical case, by adding a polynomial number of repetitions with respect to QPE.

### 5.7.2 Quantum Variational Eigensolver (QVE)

The procedure outlined above replaces the long coherent evolution required by QPE by many short coherent evolutions. In both QPE and QEE we require a good approximation to the ground-state wavefunction to compute the ground-state eigenvalue, and we now consider this problem. The quantum variational eigensolver (QVE) algorithm is a variational method to prepare the eigenstate and, by exploiting QEE, requires short coherent evolution.



**Figure 1 | Architecture of the quantum-variational eigensolver.** In QEE, quantum states that have been previously prepared are fed into the quantum modules, which compute  $\langle \mathcal{H}_i \rangle$ , where  $\mathcal{H}_i$  is any given term in the sum defining  $\mathcal{H}$ . The results are passed to the CPU, which computes  $\langle \mathcal{H} \rangle$ . In the quantum variational eigensolver, the classical minimization algorithm, run on the CPU, takes  $\langle \mathcal{H} \rangle$  and determines the new state parameters, which are then fed back to the QPU.

It is well known that the eigenvalue problem for an observable represented by an operator  $H$  can be restated as a variational problem on the RayleighRitz quotient such that the eigenvector  $|\psi\rangle$  corresponding to the lowest eigenvalue is the one that minimizes

$$\frac{\langle \psi | \mathcal{H} | \psi \rangle}{\langle \psi | \psi \rangle} \quad (5.71)$$

By varying the experimental parameters in the preparation of  $|\psi\rangle$  and computing the RayleighRitz quotient using QEE as a subroutine in a classical minimization, one may prepare unknown eigenvectors. At the termination of the algorithm, a simple prescription for the reconstruction of the eigenvector is stored in the final set of experimental parameters that define  $|\psi\rangle$ .

Note that if a quantum state is characterized by an exponentially large number of parameters, it cannot be prepared with a polynomial number of operations. The set of efficiently preparable states are therefore characterized by polynomially many parameters, and we choose a particular set of ansatz states of this type. Under these conditions, a classical search algorithm on the experimental parameters that define  $|\psi\rangle$  needs only explore a polynomial number of dimensions – requirement for the search to be efficient.

One example of a quantum state parameterized by a polynomial number of parameters for which there is no known efficient classical implementation is the

unitary coupled cluster ansatz

$$|\Psi\rangle = e^{T-T^\dagger} |\Phi\rangle_{\text{ref}} \quad (5.72)$$

where  $|\Phi\rangle_{\text{ref}}$  is some reference state, usually the Hartree Fock ground state, and  $T$  is the cluster operator for an  $N$  electron system, defined by

$$T = T_1 + T_2 + \cdots + T_N, \quad (5.73)$$

where

$$T_1 = \sum_{pr} t_p^r \hat{a}_p^\dagger \hat{a}_r, \quad T_2 = \sum_{pqrs} t_{pq}^{rs} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_r \hat{a}_s, \quad \dots \quad (5.74)$$

By construction,  $(T - T^\dagger)$  is anti-Hermitian, and exponentiation maps it to a unitary operator  $e^{(T - T^\dagger)}$ .

For any fixed excitation level  $k$ , the reduced cluster operator is written as

$$T^{(k)} = \sum_{i=1}^k T_i. \quad (5.75)$$

This state may be prepared efficiently on a quantum device. The reduced anti-hermitian cluster operator  $(T^{(k)} - T^{(k)\dagger})$  is the sum of a polynomial number of termsnamely, it contains a number of terms  $\mathcal{O}(N^k(M - N)^k)$ , where  $M$  is the number of single-particle orbitals. By defining an effective Hermitian Hamiltonian  $\mathcal{H} = i(T^{(k)} - T^{(k)\dagger})$  and performing the Jordan Wigner transformation to reach a Hamiltonian that acts on the space of qubits  $\hat{\mathcal{H}}$ , we are left with a Hamiltonian that is a sum of polynomially many products of Pauli operators. The problem then reduces to the quantum simulation of this effective Hamiltonian,  $\hat{\mathcal{H}}$  which can be one in polynomial time using known procedures.



## 5.8 Quantum Approximate Optimization Algorithm (QAOA)

## 5.9 Quantum Variational Factorization (QVF)

## **5.10 Variational Thermal Quantum Simulation via Thermofield Double States**

### **5.10.1 Thermofield Double States (TFD)**

### **5.11 Generation of TFD and Critical Ground States with a Quantum Computer**

## 5.12 Efficient variational simulation of non-trivial quantum states